

# 在 Oracle® Solaris 中安装和配置 OpenStack (Kilo)

ORACLE®

文件号码 E74917  
2016 年 6 月



文件号码 E74917

版权所有 © 2014, 2016, Oracle 和/或其附属公司。保留所有权利。

本软件和相关文档是根据许可证协议提供的，该许可证协议中规定了关于使用和公开本软件和相关文档的各种限制，并受知识产权法的保护。除非在许可证协议中明确许可或适用法律明确授权，否则不得以任何形式、任何方式使用、拷贝、复制、翻译、广播、修改、授权、传播、分发、展示、执行、发布或显示本软件和相关文档的任何部分。除非法律要求实现互操作，否则严禁对本软件进行逆向工程设计、反汇编或反编译。

此文档所含信息可能随时被修改，恕不另行通知，我们不保证该信息没有错误。如果贵方发现任何问题，请书面通知我们。

如果将本软件或相关文档交付给美国政府，或者交付给以美国政府名义获得许可证的任何机构，则适用以下注意事项：

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

本软件或硬件是为了在各种信息管理应用领域内的一般使用而开发的。它不应被应用于任何存在危险或潜在危险的应用领域，也不是为此而开发的，其中包括可能会产生人身伤害的应用领域。如果在危险应用领域内使用本软件或硬件，贵方应负责采取所有适当的防范措施，包括备份、冗余和其它确保安全使用本软件或硬件的措施。对于因在危险应用领域内使用本软件或硬件所造成的一切损失或损害，Oracle Corporation 及其附属公司概不负责。

Oracle 和 Java 是 Oracle 和/或其附属公司的注册商标。其他名称可能是各自所有者的商标。

Intel 和 Intel Xeon 是 Intel Corporation 的商标或注册商标。所有 SPARC 商标均是 SPARC International, Inc 的商标或注册商标，并按许可证的规定使用。AMD、Opteron、AMD 徽标以及 AMD Opteron 徽标是 Advanced Micro Devices 的商标或注册商标。UNIX 是 The Open Group 的注册商标。

本软件或硬件以及文档可能提供了访问第三方内容、产品和服务的方式或有关这些内容、产品和服务的信息。除非您与 Oracle 签订的相应协议另行规定，否则对于第三方内容、产品和服务，Oracle Corporation 及其附属公司明确表示不承担任何种类的保证，亦不对其承担任何责任。除非您和 Oracle 签订的相应协议另行规定，否则对于因访问或使用第三方内容、产品或服务所造成的任何损失、成本或损害，Oracle Corporation 及其附属公司概不负责。

#### 文档可访问性

有关 Oracle 对可访问性的承诺，请访问 Oracle Accessibility Program 网站 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=dacc>。

#### 获得 Oracle 支持

购买了支持服务的 Oracle 客户可通过 My Oracle Support 获得电子支持。有关信息，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>；如果您听力受损，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>。



# 目录

---

使用本文档 .....	11
<b>1 Oracle Solaris 上的 OpenStack (Kilo) 中的新增功能 .....</b>	<b>13</b>
基于核心组件的功能 .....	13
与 Cinder 相关的功能 .....	13
与 Nova 相关的功能 .....	14
与 Neutron 相关的功能 .....	14
Oracle Solaris 功能添加 .....	15
其他信息来源 .....	16
<b>2 升级到 OpenStack 2015.1.2 (Kilo) .....</b>	<b>17</b>
升级说明 .....	17
升级的过程 .....	17
▼ 如何从 Havana 升级到 Kilo .....	18
▼ 如何从 Juno 升级到 Kilo .....	18
▼ 升级后任务 .....	19
<b>3 跨多个系统安装来进行多节点 OpenStack 配置 .....</b>	<b>23</b>
三节点体系结构概述 .....	23
准备步骤 .....	25
准备主机名、变量和口令 .....	25
Keystone 脚本样例 .....	27
关于编辑配置文件 .....	27
优化内存使用 .....	28
配置 NTP 服务器 .....	28
▼ 如何设置 NTP 服务器 .....	28
配置控制器节点 .....	29
配置 NTP 客户机 .....	30
安装 MySQL .....	30
安装 Keystone .....	31

安装 Glance .....	32
安装 Nova .....	34
安装 Horizon .....	35
安装 Cinder .....	36
安装 Neutron .....	37
配置计算节点 .....	40
▼ 如何配置计算节点 .....	40
▼ 如何启用控制台访问 .....	43
配置存储节点 .....	44
▼ 如何配置块存储节点 .....	45
配置 Swift 对象存储 .....	46
▼ 如何配置 Swift 代理控制器服务节点 .....	46
▼ 如何配置对象存储节点 .....	48
4 安装后和配置任务 .....	51
为 OpenStack 项目准备外部网络 .....	51
关于提供商路由器 .....	51
创建外部网络 .....	53
为 Glance 系统信息库准备映像 .....	59
创建映像 .....	59
显示有关映像的信息 .....	61
使用 Glance 映像创建脚本 .....	62
5 使用云 .....	65
访问 OpenStack 系统信息显示板 .....	65
▼ 如何访问 OpenStack 系统信息显示板 .....	65
浏览系统信息显示板 .....	66
创建项目和用户 .....	68
▼ 如何创建项目及分配用户 .....	68
▼ 如何使用现有用户填充项目 .....	69
创建项目的内部网络 .....	70
▼ 如何为项目配置网络 .....	70
▼ 如何将浮动 IP 地址与项目相关联 .....	72
创建和引导 VM 实例 .....	73
▼ 如何创建 SSH 密钥对 .....	73
▼ 如何创建 VM 实例 .....	74
▼ 如何将用户添加到 VM 实例 .....	77
管理风格 .....	78
显示有关风格的信息 .....	79

---

修改风格规范 .....	79
管理 VM 实例 .....	82
迁移和清除 VM 实例 .....	82
调整 VM 实例大小 .....	84
<b>6 用于 Cinder 配置和部署的选项 .....</b>	<b>87</b>
部署用于存储的远程系统 .....	87
配置 cinder.conf 文件 .....	88
向指定的用户授予权限 .....	90
将远程主机启用为目标 .....	91
指定计算节点的引导卷 .....	92
▼ 如何为计算实例创建根存储卷 .....	92
使用 Cinder NFS 驱动程序 .....	93
▼ 如何使用 Cinder NFS 驱动程序 .....	93
将 OpenStack 用于 Oracle ZFS Storage Appliance .....	95
关于 Oracle ZFS Storage Appliance .....	95
使用 Oracle ZFSSA 配置 OpenStack .....	95
▼ 如何配置 Oracle ZFSSA for OpenStack .....	96
<b>7 Neutron 部署的选项 .....</b>	<b>101</b>
在内核区域中部署 Neutron .....	101
▼ 如何在内核区域中安装 Neutron 组件 .....	103
显示 MAC 地址和 VID 信息 .....	104
从来宾 VM 内显示 .....	104
从主机显示 .....	104
<b>8 使用 Ironic .....</b>	<b>107</b>
关于 Ironic 组件 .....	107
安装和配置 Ironic .....	108
▼ 如何安装和配置 Ironic .....	108
概述：包含 Ironic 的裸机部署 .....	112
使用 Ironic 部署裸机 .....	114
▼ 如何通过 UAR 文件部署裸机 .....	114
▼ 如何停用节点 .....	117
<b>9 使用 Heat .....</b>	<b>119</b>
关于 Heat 组件 .....	119
安装 Heat .....	119

▼ 如何配置 Heat .....	120
关于 HOT 模板 .....	120
将 Heat 与 Cloudbase-Init 一起使用 .....	121
▼ 如何自动初始化来宾映像 .....	122
<b>10 OpenStack 故障排除 .....</b>	<b>125</b>
获取命令行帮助 .....	125
已知限制 .....	126
检查日志文件 .....	127
调查和解决问题 .....	129
网络创建 .....	130
VM 实例安装和配置 .....	130
与 Horizon 相关的问题 .....	132
可伸缩性问题 .....	133
中断网络 .....	133
▼ 如何删除 Neutron 上的网络配置 .....	134
▼ 如何删除 Vport .....	134
有关调试的一般提示和技巧 .....	135
有用的站点 .....	135
<b>A 常用 OpenStack 配置文件和服务 .....</b>	<b>137</b>
配置文件 .....	137
Cinder 文件 .....	137
Glance 文件 .....	137
Keystone 文件 .....	137
Neutron 文件 .....	137
Nova 文件 .....	138
Horizon 文件 .....	138
Swift 文件 .....	138
OpenStack SMF 服务 .....	138
Cinder .....	138
Glance .....	139
Keystone .....	139
Neutron .....	139
Nova * .....	139
Swift .....	139
<b>B OpenStackClient 命令 .....</b>	<b>141</b>
OpenStackClient (OSC) .....	141



索引 ..... 143



## 使用本文档

---

- 概述 - 介绍如何在支持的 Oracle Solaris 系统上安装当前 OpenStack 版本和部署 OpenStack 虚拟机。
- 目标读者 - 大型安装系统管理员。
- 必备知识 - Oracle Solaris 网络和大型系统管理。熟悉 OpenStack 会很有用。

## 产品文档库

有关该产品及相关产品的文档和资源，可从以下网址获得：<http://www.oracle.com/pls/topic/lookup?ctx=E69403>。

## 反馈

可以在 <http://www.oracle.com/goto/docfeedback> 上提供有关本文档的反馈。



# Oracle Solaris 上的 OpenStack (Kilo) 中的新增功能

---

Oracle Solaris 11.3 SRU 9 在其整体软件包中包括了 OpenStack 2015.1.2 (Kilo) 版本。本章列出了此发行版中 OpenStack 的 Kilo 版本的新增功能。

- “[基于核心组件的功能](#)” [13]
- “[其他信息来源](#)” [16]

## 基于核心组件的功能

本节介绍引入到 OpenStack 的 Kilo 版本核心组件的功能。

## 与 Cinder 相关的功能

增加了用于 Cinder 的以下功能：

- 远程 SAN 存储的使用  
对存储区域网络 (storage area network, SAN) 的支持允许您远程部署 Cinder 服务。有关更多信息，请参见[“部署用于存储的远程系统”](#) [87]。
- 支持卷的备份和恢复操作  
Oracle Solaris 中现在启用了 Cinder 备份 SMF 服务。因此，现在可以在所配置的后端之间备份和恢复未附加的卷。当前，Swift 是唯一受支持的后端。
- 支持 Cinder 卷的迁移  
Cinder 利用 ZFS 操作来迁移 Cinder 卷。ZFS 发送和接收进程使得卷能够在所配置的所有 Cinder 后端之间迁移。如果迁移目的地与源位于同一池上，则会使用 ZFS 重命名操作。当前，cinder 卷迁移支持仅限于单个系统。
- 通过 *manage* 选项增强了 Cinder 中的卷管理，以便可以导入在 Cinder 功能之外创建的卷。导入这些卷后，可以像常规 Cinder 卷那样在云中管理它们。  
同样，通过 *unmanage* 选项可以使 Cinder 卷变为不可见以禁止访问。该选项并不删除卷。这样，就可以通过重新导入这些卷来重新启用对它们的访问。

管理/取消管理功能在 Horizon 系统信息显示板以及命令行上均可用。

- 在当前的 Kilo 实现中提供了具有新属性的已更新 ZFSSA Cinder 驱动程序。要在 OpenStack 中调整 ZFSSA 配置，请参见[升级后任务 \[19\]](#)了解相应说明。
- 提供了对 Solaris 的 OpenStack Cinder NFS 卷驱动程序支持。可以创建类型为 `nfs` 的卷。NFS 文件访问在 Cinder 中定义为用户和组。但是，请注意，当前此驱动程序的支持限于内核区域。  
有关更多信息，请参见[“使用 Cinder NFS 驱动程序” \[93\]](#)。
- 在具有多个后端的配置中，可以为您创建的每个计算节点指定引导卷。要通过此功能配置 Cinder，请参见[“指定计算节点的引导卷” \[92\]](#)。

## 与 Nova 相关的功能

增加了用于 Nova 的以下功能：

- 安全实时迁移  
对实时迁移（Oracle Solaris 区域的一个功能）的支持已扩展到 Nova 节点中的 VM 实例。有关节点实时迁移的更多信息，请参见[“迁移和清除 VM 实例” \[82\]](#)。
- 支持实例清除  
如果出现主机故障或者服务在主机上变为禁用状态，则可以通过使用 `nova evacuate` 命令将实例移动到其他节点以进行恢复。请注意，仅当根设备在共享存储上时，才支持清除。而且，仅内核区域支持清除，非全局区域不支持。
- 调整 VM 实例大小的功能  
通过更改 VM 实例的风格可以调整其大小。新风格提供具有不同属性（如 CPU 容量、内存和其他资源）的 VM 实例。有关更多信息，请参见[“调整 VM 实例大小” \[84\]](#)。

## 与 Neutron 相关的功能

增加了用于 Neutron 的以下功能：

- 内核区域中 Neutron 的功能  
对区域中动态 MAC 地址和 VID 的支持允许您在内核区域中安装 Neutron。有关更多信息，请参见[“在内核区域中部署 Neutron” \[101\]](#)。
- VPN 作为服务  
通过 Neutron 可支持 VPN 即服务 (VPN as a service, VPNaaS)。此外，“网络 IPsec 管理”配置文件添加到已分配给 Neutron 的配置文件。此配置文件允许管理员管理 IPsec 和 IKE 系统管理工具 (system management facility, SMF) 服务。

## Oracle Solaris 功能添加

这些功能添加是 Oracle Solaris 的 OpenStack 驱动程序端的增强功能。在核心上游项目中已实现这些增强功能。

### 对 cloudbase-init 的支持

cloudbase-init 服务可促进云中客操作系统的初始化和配置。这些任务包括用户创建、口令生成、静态网络配置、主机名、SSH 公钥和用户数据脚本。服务的配置文件是 `/etc/cloudbase-init.conf`。

cloudbase-init 的 Oracle Solaris 版本作为 SMF 服务 `application/cloudbase-init` 运行，缺省情况下会启用它。通过用户数据导出的脚本通常执行需要特权访问的系统 and 应用程序配置任务。因此，cloudbase-init 服务将以 root 用户身份运行，并且任何用户数据脚本也将以 root 用户身份运行。

cloudbase-Init 软件包未包括在任何标准组软件包中。用户应仅将软件包安装到明确要部署在云环境中的映像中。

---

注 - 当前在此 OpenStack 发行版中，`/etc/cloudbase-init.conf` 文件仅启用 UserData 插件。

---

有关 Cloudbase-Init 的更多信息，请转到 <http://cloudbase-init.readthedocs.io/en/latest/tutorial.html>。

### OpenStackClient 的实现

OpenStackClient (OSC) 是来自 OpenStack 社区的客户机，它将所有组件命令集组合到具有统一命令结构的单个 shell 中。因此，如果以前的版本具有基于组件的命令，如 `keystone user-list`、`glance image-show`、`neutron net-list` 等，则发出其中的大多数命令时以 `openstack` 作为主命令，如 `openstack user list`。

在当前的 Kilo 版本中，所有的 keystone 命令都已过时。如果使用 keystone 命令，则将生成相应的警报。

有关 OSC 的更多信息，请参见 <http://docs.openstack.org/developer/python-openstackclient/index.html>。

有关以前的命令及其在 OSC 中等效项的列表，请参见[附录 B, OpenStackClient 命令](#)。

## 其他信息来源

另请参见 OpenStack 社区的 Kilo 发行说明所提供的信息，网址为 <https://wiki.openstack.org/wiki/ReleaseNotes/Kilo>：

有关此 OpenStack 发行版的当前 Oracle Solaris 实现的问题列表，请参见 [My Oracle Support \(https://support.oracle.com\)](https://support.oracle.com) 中 OpenStack 版本的对应自述文件。



## 升级到 OpenStack 2015.1.2 (Kilo)

---

本章介绍从以前版本升级到 OpenStack 的 Kilo 版本的步骤。

- [“升级说明” \[17\]](#)
- [“升级的过程” \[17\]](#)

### 升级说明

在升级时，请考虑以下最佳做法：

- 创建备份，尤其是仅升级 OpenStack 配置而不是整个操作系统时。
- 如果仅升级 OpenStack 配置，则随升级的 OpenStack 版本创建新的引导环境 (boot environment, BE)。升级完成时，引导到新的 BE。
- 在多节点配置中，首先更新控制器节点，然后更新存储节点，最后更新其余节点。
- 在 Kilo 中，会禁用 RabbitMQ 的缺省来宾用户和口令设置。如果在以前的 OpenStack 版本中使用了缺省口令设置，请更改为安全口令。

### 升级的过程

升级到 Kilo 的步骤取决于从其开始的 OpenStack 版本。不支持从 Havana 直接升级到 Kilo。如果当前版本是 Havana，则需要先升级到 Juno，然后执行从 Juno 到 Kilo 的另一个升级。

---

注 - 执行升级时，必须使用 `pkg update` 命令启动操作。不要使用 `pkg install`。

---

升级任务列表如下所示：

- 从现有的 Havana 配置，从[如何从 Havana 升级到 Kilo](#)开始。然后继续执行[升级后任务](#)。
- 从现有的 Juno 配置，从[如何从 Juno 升级到 Kilo](#)开始。然后继续执行[升级后任务](#)。

## ▼ 如何从 Havana 升级到 Kilo

如果当前的 OpenStack 版本是 Havana（它正在 Oracle Solaris 11.2 SRU 11 或以前发行版上运行），请使用此过程。

如果 OpenStack 配置在多个节点上构建，则必须在每个节点上执行此过程，从控制器开始。

Oracle Solaris 升级将为操作系统创建新的引导环境 (boot environment, BE)。完成升级后，将激活新的 BE。重新引导时，系统将引导到新的 BE。

1. 如果系统运行的是 Oracle Solaris 11.2 SRU 11 或以前的发行版，请执行以下步骤：
  - a. 将操作系统至少升级到 Oracle Solaris 11.3。

---

注 - 此时，尚不要升级到 Oracle Solaris 11.3 SRU 9。

---

Oracle Solaris 11.3 软件包中包括 OpenStack Juno 软件包。

有关升级到 Oracle Solaris 11.3 的步骤，请参阅《[Updating to Oracle Solaris 11.3](#)》中的“[How to Update a System Running 11.1 or 11.2 to Oracle Solaris 11.3](#)”。

- b. 通过检查来确定有关迁移 Neutron 数据库的信息是否适用于您的设置。

请参见 [Migrating Neutron Database from sqlite to MySQL for Oracle OpenStack for Oracle Solaris](#)（针对 Oracle OpenStack for Oracle Solaris 将 Neutron 数据库从 sqlite 迁移到 MySQL）。如果方案适用于您的配置，请按照博客条目中的说明操作。
2. 将操作系统升级到 Oracle Solaris 11.3 SRU 9。

有关如何升级到 Oracle Solaris 11.3 SRU 9 发行版的说明，请登录到 <https://support.oracle.com> 中的 MOS 帐户。从 Oracle Solaris 11.3 支持系统信息库更新 (Support Repository Update, SRU) 索引（文档 ID 2045311.1），访问 SRU9 自述文件。
3. 将系统引导到新的 BE。

接下来的步骤 通过完成[升级后任务 \[19\]](#)中的所有步骤，完成升级过程。

## ▼ 如何从 Juno 升级到 Kilo

如果当前的 OpenStack 版本是 Juno，请使用此过程。

如果 OpenStack 配置在多个节点上构建，则必须在每个节点上执行此过程，从控制器开始。

Oracle Solaris 升级将为操作系统创建新的引导环境 (boot environment, BE)。完成升级后，将激活新的 BE。重新引导时，系统将引导到新的 BE。

1. 将操作系统升级到 Oracle Solaris 11.3 SRU 9。  
有关如何升级到 Oracle Solaris 11.3 SRU 9 发行版的说明，请参阅该发行版的自述文件。
2. 继续执行[升级后任务](#)以完成升级过程。

## ▼ 升级后任务

升级到当前 Oracle Solaris 发行版后，执行这些剩余的步骤以完成到 OpenStack Kilo 的升级。

1. 将 Horizon 定制迁移到 Kilo 版本。
  - a. 将定制设置从 `/etc/openstack_dashboard/local_settings.py.old` 传输到 `/etc/openstack_dashboard/local_settings.py`。
  - b. 如果以下内容之一适用于您的配置，请注释掉 `/etc/openstack_dashboard/local_settings.py` 文件中的其他行：
    - 具有单节点 OpenStack 配置以达到评估目的。
    - Horizon 配置未使用 SSL。

请参见以下示例：

```
# SECURE_PROXY_SSL_HEADER = ('HTTP_X_FORWARDED_PROTOCOL', 'https')
# CSRF_COOKIE_SECURE = True
# SESSION_COOKIE_SECURE = True
```

- c. 将 Horizon Apache 配置段样例复制到 `Apache conf.d` 目录。  
必须仅复制与您所使用的协议相对应的段样例。发出以下命令之一：
  - 如果正在使用 HTTP：
 

```
# cp /etc/apache2/2.4/samples-conf.d/openstack-dashboard-http.conf /etc/apache2/2.4/conf.d
```
  - 如果正在使用 TLS：

```
# cp /etc/apache2/2.4/samples-conf.d/openstack-dashboard-tls.conf /etc/
apache2/2.4/conf.d
```

2. 如果具有多节点配置，则使用以粗体显示的行更新 `/etc/rabbitmq/rabbitmq.config`：

```
% FHC read buffer has been disabled by default in later versions of
%RabbitMQ.
[
  {rabbit, [
    {fhc_read_buffering, false},
    {loopback_users, []}
  ]}
].
```

3. 更新 Cinder v2 服务。

在 Keystone 正运行的节点上执行这些步骤。对于发出的每个命令，都包括输出样例。

- a. 创建 v2 Cinder 服务。

```
controller# openstack --os-url http://$CONTROLLER_ADMIN_NODE:35357/v2.0 \
--os-token ADMIN \
  service create --name cinderv2 \
  --description "Cinder Volume Service v2" volumev2
```

Field	Value
description	Cinder Volume Service v2
enabled	True
id	2ee6fefbdcdc4f06bcb0e36e0e4dd9c3
name	cinderv2
type	volumev2

- b. 创建端点。

```
controller# openstack --os-url http://$CONTROLLER_ADMIN_NODE:35357/v2.0 \
--os-token ADMIN
  endpoint create \
  --region RegionOne \
  --publicurl "http://$CONTROLLER_ADMIN_NODE:8776/v2/\$(tenant_id)s" \
  --adminurl "http://$CONTROLLER_ADMIN_NODE:8776/v2/\$(tenant_id)s" \
  --internalurl "http://$CONTROLLER_ADMIN_NODE:8776/v2/\$(tenant_id)s" cinderv2
```

Field	Value
adminurl	http://controller-node:8776/v2/\\$(tenant_id)s
id	1b8cd962b12342429cdedb0c7e5d0440
internalurl	http://controller-node:8776/v2/\\$(tenant_id)s
publicurl	http://controller-node:8776/v2/\\$(tenant_id)s
region	RegionOne

```

| service_id | 2ee6fefbdcdc4f06bcb0e36e0e4dd9c3 |
| service_name | cinderv2 |
| service_type | volumev2 |
+-----+

```

c. 验证 cinderv2 是否存在于端点列表中：

```

controller# openstack --os-url http://$CONTROLLER_ADMIN_NODE:35357/v2.0 --os-token
ADMIN endpoint list
+-----+-----+-----+-----+
| ID | Region | Service Name | Service Type |
+-----+-----+-----+-----+
| 6891354066f84268968c8498f5f6d51b | RegionOne | neutron | network |
| 03121908d41e4efa98748fde8ca6d057 | RegionOne | heat | orchestration |
| b69e4f0373ff4a8f9560dc2644d891ba | RegionOne | glance | image |
| 1e6c7f52dc34a27b7ccac98918f19f1 | RegionOne | ec2 | ec2 |
| e3236915a3dd4098b9e8e0853b5a5af2 | RegionOne | keystone | identity |
| fe8870c3e6ac4b529aa7ce7563fc24a4 | RegionOne | heat-cfn | cloudformation |
| aa931a795f2c4c0ca637e0e4c351cf07 | RegionOne | swift | object-store |
| 1b8cd962b12342429cdedb0c7e5d0440 | RegionOne | cinderv2 | volumev2 |
| 618a8edba487417c91d0de7f3bcc786d | RegionOne | cinder | volume |
| 4c79d020189a44d383bdc15033a942c4 | RegionOne | nova | compute |
+-----+-----+-----+-----+

```

4. 重新启动 Apache 服务。

```
# svcadm restart apache24
```

5. 如果 IP 过滤器服务未运行，请启动它。

```
controller# svcadm enable -rs ipfilter
```

6. 如果正在将 ZFSSA 用于存储，请调整 `/etc/cinder.cinder.conf` 以使用新的驱动程序属性。

zfssa\_initiator\_config 属性列出了多个启动器或多组启动器，并替换 zfssa\_initiator\_group（它在 OpenStack Kilo 版本中已过时）。

a. 通过使用以下格式，列出新属性的多个启动器：

```

zfssa_initiator_config = {
  'init-grp1': [
    {'iqn': 'iqn1', 'user': 'user', 'password': 'password'},
    {'iqn': 'iqn2', 'user': 'user', 'password': 'password'}
  ],
  'init-grp2': [
    {'iqn': 'iqn3', 'user': 'user', 'password': 'password'}
  ]
}

```

例如，如果在 ZFS 存储设备上创建了两组启动器（A 组和 B 组），请对它们进行如下定义：

```
zfssa_initiator_config = {
```

```
    'GroupA':[
      {'iqn':'iqn.1986-03.com.sun:01:0a43b9fdcfd5.570d7fd1', 'user':'test1',
'password':'password1234'},
      {'iqn':'iqn.1986-03.com.sun:01:0a43b9fdcfd5.570d7fd2', 'user':'',
'password':''}
    ],
    'GroupB':[
      {'iqn':'iqn.1986-03.com.sun:01:0a43b9fdcfd5.570d7fd3', 'user':'',
'password':''}
    ]
  }
```

b. 注释掉文件中的以下已过时参数：

- `zfssa_initiator_group`
- `zfssa_initiator`

c. 重新启动 Cinder 服务。

```
controller# svcadm restart cinder-volume:default
```

## 跨多个系统安装来进行多节点 OpenStack 配置

---

本章介绍如何安装多节点 OpenStack 配置。本章包括以下几个主题：

- “三节点体系结构概述” [23]
- “准备步骤” [25]
- “配置控制器节点” [29]
- “配置计算节点” [40]
- “配置存储节点” [44]
- “配置 Swift 对象存储” [46]

### 三节点体系结构概述

单节点配置对于将 OpenStack 作为产品进行测试以及熟悉其功能非常有用。但是，单节点配置不适合在生产环境中使用。对于此环境，您将在多个系统或节点中安装并配置 OpenStack。

每个云只需要一个系统信息显示板实例、一个映像存储和一个身份服务。每个云可以具有任意数量的存储和计算实例。根据您的特定云部署的需求评估各个组件，以确定该组件是否应该安装在单独的节点上，以及所需的该类型节点的数量。

本章中介绍的体系结构在以下三个系统中部署：

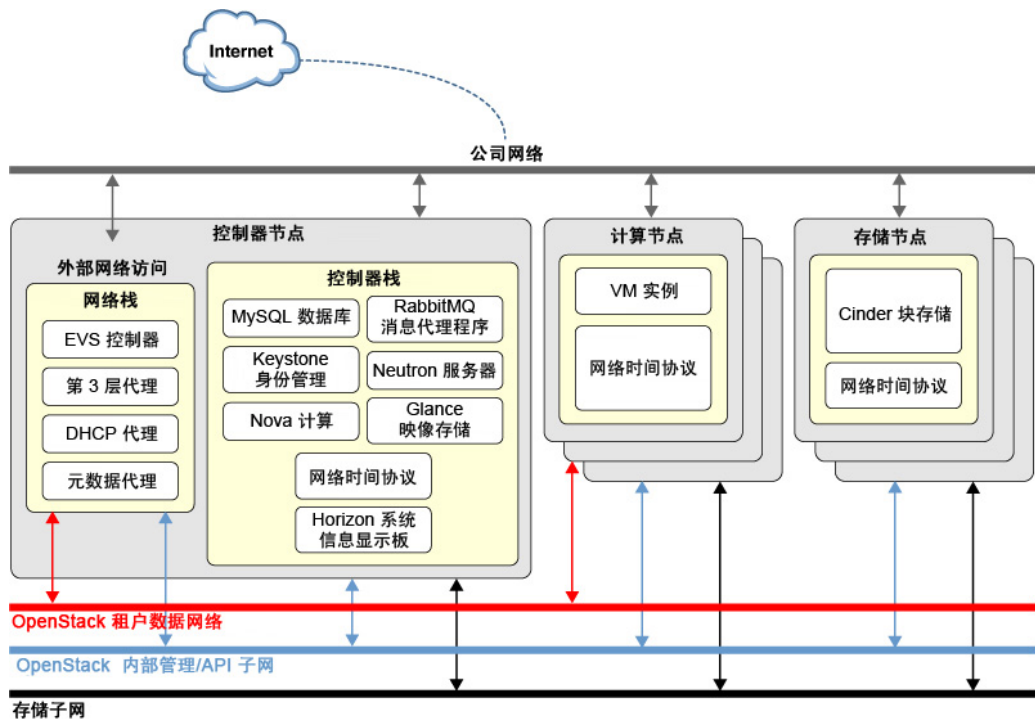
- 控制器节点 - 运行大多数共享 OpenStack 服务以及其他工具的节点。控制器节点为云提供 API、调度和其他共享服务。控制器节点具有系统信息显示板、映像存储以及身份服务。另外，还在此节点中配置 Nova 计算管理服务以及 Neutron 服务器。
- 计算节点 - 安装 VM 实例（也称为 Nova 计算实例）的节点。该节点运行管理这些 VM 实例的计算守护进程。
- 存储节点 - 托管数据的节点。

此三节点体系结构仅是在多个系统上部署 OpenStack 的一种方式。因为其灵活性，您可以除了此体系结构之外的其他方式分发 OpenStack 组件。因此，您必须在开始安装之前规划云配置。有关引导您的规划的信息，请参阅《[OpenStack 配置规划](#)》。

注 - 要为单个 Oracle SPARC 服务器分区并在运行 OVM Server for SPARC (LDoms) 的服务器上配置多节点 OpenStack，请参见《[Multi-node Solaris 11.2 OpenStack on SPARC Servers](#)》（《SPARC 服务器上的多节点 Solaris 11.2 OpenStack》）。该文专门提到 Havana 版本的 OpenStack。但是，一般步骤也适用于当前版本。

下图显示了本章中介绍的体系结构的概括性视图。

图 1 三节点配置参考体系结构



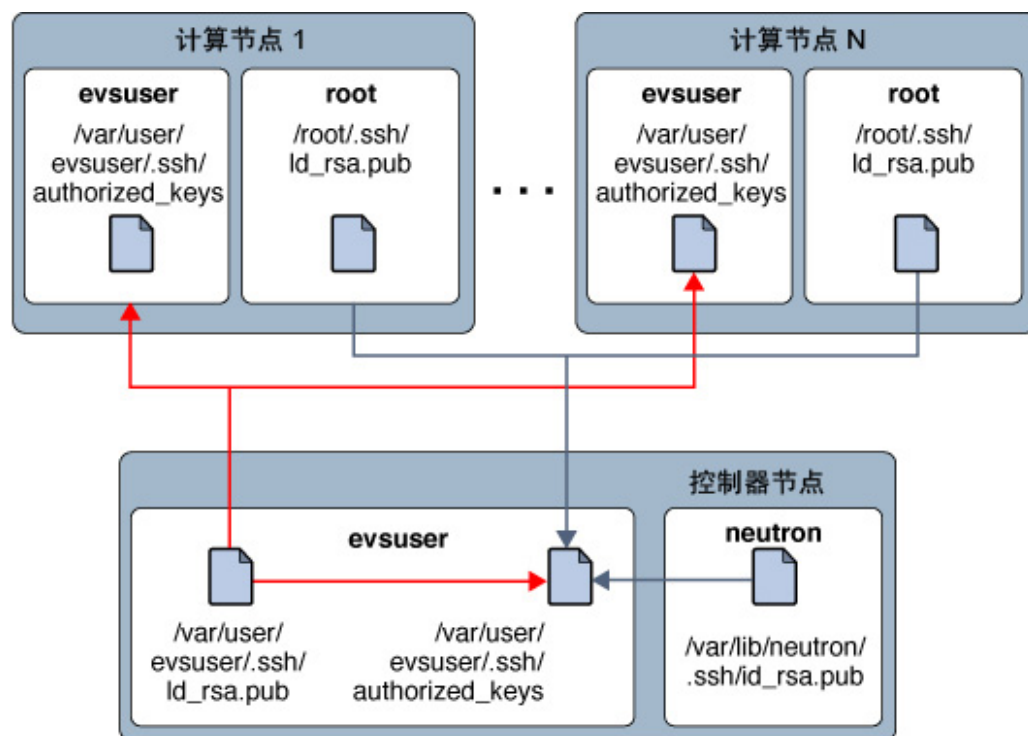
此图使用了 Cinder 来展示存储节点。不过，可以对 Swift 对象存储服务进行类似配置。

在 Oracle Solaris 中，弹性虚拟交换机 (elastic virtual switch, EVS) 形成 OpenStack 网络的后端。EVS 可促进 VLAN 或 VXLAN 上的 VM 实例之间的通信。这些 VM 实例可以位于同一计算节点上，也可以位于多个计算节点上。有关 EVS 的更多信息，请参阅关于弹性虚拟交换机的《管理网络虚拟化和网络资源》。此书位于 [Operating Systems Documentation](#) (操作系统文档) 中您 Oracle Solaris 版本相应的库中。

要让不同节点相互通信，控制器节点中的 `evsuser`、`neutron` 和 `root` 的 SSH 公钥必须包含在所有配置的计算节点中的每个 `evsuser` 的 `authorized_keys` 文件中。请参阅下图，其中显示 SSH 公钥的分配。该图假设配置了多个计算节点。



图 2 EVS 控制器 SSH 密钥分配



有关适用于 Oracle Solaris 系统上 OpenStack 部署的 OpenStack 配置参数列表，请参见附录 A, 常用 OpenStack 配置文件和服务。

## 准备步骤

本节讨论了在实施多节点 OpenStack 配置之前的一些准备工作注意事项。

### 准备主机名、变量和口令

在多节点配置中，将使用多个网络接口来为云创建的不同子网提供服务。请确保您已为这些接口准备了主机名。请在系统的 `/etc/hosts` 文件或 DNS 配置中包括这些名称及其 IP 地址。

例如，您可能要创建以下主机名来处理不同类型的网络通信。

- 用于 OpenStack 网络的 host-on，该网络承载着管理和 API 通信。
- 用于项目网络的 host-tn，该网络承载着计算节点与 L3 路由器之间的通信。
- 用于外部网络通信的 host-en。

在不同的节点中配置 OpenStack 服务时，请创建变量以便于执行任务，例如以下示例：

- `$CONTROLLER_ADMIN_NODE` – OpenStack 管理服务连接到的控制器节点中的接口主机名称或 IP 地址。
- `$CONTROLLER_ADMIN_NODE_IP` – 处理 OpenStack 管理服务和通信的控制器端口的 IP 地址。
- `$COMPUTE_ADMIN_NODE_IP` – 处理 OpenStack 管理服务和通信的计算端口的 IP 地址。
- `$VOLUME_IP` – 控制器节点的主机名。

口令在配置过程中也是必需的。下面是需要准备的口令的样例列表。

- MySQL 数据库的 Root 口令
- 用户 admin 的口令
- OpenStack 服务的数据库口令：
  - 身份服务
  - 映像服务
  - 计算服务
  - 系统信息显示板数据库
  - 块存储数据库
  - 网络数据库
  - Orchestration 数据库
- OpenStack 服务用户的口令：
  - glance
  - nova
  - cinder
  - neutron
  - heat

---

注 - 还可以为一组用户或服务分配一个共同的口令。无论采用哪个系统分配口令，请确保按照最佳做法来保护环境。请参见 [Operating Systems Documentation](#)（操作系统文档）中您 Oracle Solaris 版本相应库中的《确保系统和连接设备的安全》。

---

## Keystone 脚本样例

要快速填充 Keystone 数据库，可使用样例脚本 `/usr/demo/openstack/keystone/sample_data.sh`。该脚本执行以下基本任务来帮助您入手：

- 创建以下初始项目：
  - `service`，在其下创建基本或核心服务。
  - `demo`，将在其下创建使用 `secrete` 作为缺省口令的用户 `admin`。
- 填充 Keystone 数据库。
- 创建以下核心服务。
  - `cinder`
  - `cinder2`
  - `ec2`
  - `glance`
  - `keystone`
  - `neutron`
  - `nova`
  - `swift`
  - `heat`

所有服务都具有对应的用户名和口令，但是 Keystone 服务（不为其创建用户）除外。缺省情况下，用户名、口令和服务名称是相同的。例如，`cinder` 是 Cinder 服务的用户名和口令，`glance` 是 Glance 服务的用户名和口令，以此类推。作为一个选项，可以创建定制的口令以替换脚本中的这些缺省口令。或者，也可以为脚本中的所有服务设置一个口令。向脚本应用所有必需的更改，然后运行该脚本以引导 Keystone。

---

注 - 请确保查看该脚本，以了解有关可为环境设置的参数的更多信息。根据偏好替换脚本中的缺省设置。

在本文档中，过程假定：除口令之外，在使用样例数据脚本的过程中没有进行修订，并且所有脚本的缺省设置都应用到云配置中。

---

## 关于编辑配置文件

OpenStack 配置的主要部分涉及编辑组件的配置文件。在本文档中，仅标识了选定的参数以在各 `*.conf` 或 `*.ini` 文件中进行配置。这些选定的参数是使云配置起作用所必需的最低要求。但是，应查看各配置文件的整个内容，以确保与特定云设置相关的所有参数均正确配置。

## 优化内存使用

为更好地管理 Oracle Solaris 11 中 ZFS 与应用程序之间的内存使用情况，请在该节点上设置 `usr_reserve_hint_pct` 参数，如以下示例所示：

```
# echo "set user_reserve_hint_pct=80" >>/etc/system.d/site:kernel-zones-reserve
# reboot
```

其中 `site` 是指您的公司。

还要在不同的 OpenStack 节点上设置此参数。

有关此参数的更多信息，请在 <https://support.oracle.com> 中登录到您的 MOS 帐户，然后查看文档 1663862.1 《*Memory Management Between ZFS and Applications in Oracle Solaris 11.2*》。

## 配置 NTP 服务器

虽然安装网络时间协议 (Network Time Protocol, NTP) 为可选步骤，但是强烈建议执行此步骤。NTP 帮助确保云中所有的服务节点采用一致的时间。如果在网络中启用 NTP，则配置服务节点以通过网络获取其时间。

- 如果在服务节点所在的 IP 子网上启用了 IP 多播功能，则可以利用 IP 多播功能配置 NTP。
- 如果在服务节点所在的 IP 子网上没有启用 IP 多播功能，则手动配置 NTP。

使用 NTP 需要配置 NTP 服务器和 NTP 客户机。通常，NTP 服务器是一个单独的系统，不同于其他配置 OpenStack 的系统。NTP 客户机是在承载着 OpenStack 组件的节点或系统上安装并配置的。

有关 NTP 的更多信息，请参阅位于以下网址的文档：<http://www.ntp.org/documentation.html>。

### ▼ 如何设置 NTP 服务器

NTP 服务器位于与 OpenStack 节点不同的系统上。

1. 安装 NTP 软件包。

```
ntp-server# pkg install ntp
```

2. 安装配置文件。

```
ntp-server# cp /etc/inet/ntp.server /etc/inet/ntp.conf
```

3. 编辑 `/etc/inet/ntp.conf` 文件：配置 `server` 和 `driftfile` 关键字。  
例如：

```
server 127.127.1.0 prefer
...
driftfile /var/ntp/ntp.drift
```

---

注 - 127.127.1.0 不是 IP 地址。它是一种格式，用来引用向服务器提供准确时间的时钟。请务必阅读 `ntp.conf` 文件中对 `server` 关键字进行解释的注释。

---

4. 根据您上一步骤中的定义创建 `/var/ntp/ntp.drift` 文件。

```
ntp-server# touch /var/ntp/ntp.drift
```

5. 启动 `ntp` 服务。

```
ntp-server# svcadm enable ntp
```

## 配置控制器节点

控制器节点具有一个系统信息显示板服务、一个映像存储和一个身份服务。此节点还包括 MySQL、RabbitMQ 以及计算、块存储和网络服务。

要配置控制器节点，请使用以下命令在系统上安装 OpenStack 组件和服务：

```
controller# pkg install openstack
```

在软件包安装完成后，配置您要在节点上运行的服务。以下列表指定用于配置控制器节点的任务：

- “配置 NTP 客户机” [30]
- “安装 MySQL” [30]
- “安装 Keystone” [31]
- “安装 Glance” [32]
- “安装 Nova” [34]
- “安装 Horizon” [35]
- “安装 Cinder” [36]
- “安装 Neutron” [37]

## 配置 NTP 客户机

在云部署中的每个服务节点上安装 NTP 客户机服务。

### ▼ 如何配置 NTP 客户机

本节假设您已如[如何设置 NTP 服务器 \[28\]](#)中所述设置了 NTP 服务器。

1. 创建客户机配置文件。

```
controller# cp /etc/inet/ntp.client /etc/inet/ntp.conf
```

2. 在客户机配置文件中，取消注释一个或多个服务器选项并提供 NTP 服务器的特定名称或 IP 地址。

例如，如果您配置的 NTP 服务器的主机名为 `system1`，则配置文件将如下例所示：

```
# multicastclient 224.0.1.1
...
server system1.example.com iburst
# server server_name2 iburst
# server server_name3 iburst
```

3. 启用 ntp 服务。

```
controller# svcadm enable ntp
```

## 安装 MySQL

许多 OpenStack 服务维护一个数据库来跟踪重要资源、使用情况和其他信息。尤其是对于多节点配置，请使用数据库（如 MySQL 数据库）存储此信息。

### ▼ 如何安装 MySQL 数据库

1. 启用 RabbitMQ 服务。

```
controller# svcadm enable rabbitmq
controller# svcadm restart rad:local
```

2. （可选）如果要专用 IP 地址用于管理和 API 通信，并且在 `/etc/mysql/5.5/my.cnf` 中添加该地址：

```
bind-address=${CONTROLLER_ADMIN_NODE_IP}
```

### 3. 启用 MySQL 服务。

```
controller# svcadm enable mysql
```

### 4. 设置 MySQL 服务器 root 口令。

```
controller# mysqladmin -u root password MySQL-root-password
```

### 5. 配置 MySQL。

创建 OpenStack 将使用的表。向控制器节点上的服务授予特权，以便提供对于这些数据库的独占访问。

```
controller# mysql -u root -p
Enter password: MySQL-root-password
mysql> drop database if exists nova;
mysql> drop database if exists cinder;
mysql> drop database if exists glance;
mysql> drop database if exists keystone;
mysql> drop database if exists neutron;
mysql> drop database if exists heat;
mysql> create database cinder default character set utf8 default collate utf8_general_ci;
mysql> grant all privileges on cinder.* to 'cinder'@'$CONTROLLER_ADMIN_NODE' identified
  by 'service-password';
mysql> create database glance default character set utf8 default collate utf8_general_ci;
mysql> grant all privileges on glance.* to 'glance'@'$CONTROLLER_ADMIN_NODE' identified
  by 'service-password';
mysql> create database keystone default character set utf8 default collate
  utf8_general_ci;
mysql> grant all privileges on keystone.* to 'keystone'@'$CONTROLLER_ADMIN_NODE'
  identified by 'service-password';
mysql> create database nova default character set utf8 default collate utf8_general_ci;
mysql> grant all privileges on nova.* to 'nova'@'$CONTROLLER_ADMIN_NODE' identified by
  'service-password';
mysql> create database neutron default character set utf8 default collate
  utf8_general_ci;
mysql> grant all privileges on neutron.* to 'neutron'@'$CONTROLLER_ADMIN_NODE' identified
  by 'service-password';
mysql> create database heat default character set utf8 default collate utf8_general_ci;
mysql> grant all privileges on heat.* to 'heat'@'$CONTROLLER_ADMIN_NODE' identified by
  'service-password';
mysql> flush privileges;
mysql> quit
```

## 安装 Keystone

Keystone 服务应该在控制器节点上安装并配置。此过程使用[“Keystone 脚本样例” \[27\]](#)中所述的样例脚本。在使用脚本前先阅读该节内容。

## ▼ 如何安装并配置 Keystone

1. 为 Keystone 及其他 OpenStack 服务创建共享令牌。

该令牌由随机字符串组成。请注意，openssl 命令会提示您输入构成密钥的组成部分，例如国家/地区、省/市/自治区，等等。

```
controller# openssl rand -hex 10
token-string
```

2. 将令牌设置为 shell 变量。

```
controller# export MY_SERVICE_TOKEN=token-string
```

其中，*token-string* 是来自上一步的命令的输出。

3. 在 `/etc/keystone/keystone.conf` 文件中修改参数。

配置应类似以下示例。

```
[DEFAULT]
admin_token = token-string
...
[database]
connection = mysql://keystone:service-password@$CONTROLLER_ADMIN_NODE/keystone

[oslo_messaging_rabbit]
rabbit_host=$CONTROLLER_ADMIN_NODE
```

4. 启用 Keystone SMF 服务。

```
controller# svcadm enable keystone
```

5. 使用 Keystone 样例脚本填充 Keystone 数据库。

确保运行脚本前先根据您的偏好查看并修改了脚本。过程假定样例脚本未进行定制。

```
controller# CONTROLLER_PUBLIC_ADDRESS=$CONTROLLER_ADMIN_NODE \
CONTROLLER_ADMIN_ADDRESS=$CONTROLLER_ADMIN_NODE \
CONTROLLER_INTERNAL_ADDRESS=$CONTROLLER_ADMIN_NODE \
SERVICE_TOKEN=$MY_SERVICE_TOKEN \
ADMIN_PASSWORD=admin-password \
SERVICE_PASSWORD=service-password \
/usr/demo/openstack/keystone/sample_data.sh
```

## 安装 Glance

设置 Glance 时，需要配置一些用于验证的信息，并指定 MySQL 和 RabbitMQ 服务的位置。



## ▼ 如何安装并配置 Glance

### 1. 通过在以下配置文件中取消注释或设置参数来配置 Glance :

#### ■ /etc/glance/glance-api.conf

```
[DEFAULT]
registry_host = $CONTROLLER_ADMIN_NODE

auth_strategy = keystone
default_publisher_id = image.$CONTROLLER_ADMIN_NODE

[database]
connection = mysql://glance:service-password@$CONTROLLER_ADMIN_NODE/glance

[keystone_auth token]
auth_uri = http://$CONTROLLER_ADMIN_NODE:5000/v2.0
identity_uri = http://$CONTROLLER_ADMIN_NODE:35357
admin_user = glance
admin_password = service-password
admin_tenant_name = service

[oslo_messaging_rabbit]
rabbit_host = $CONTROLLER_ADMIN_NODE
```

#### ■ /etc/glance/glance-cache.conf

```
[DEFAULT]
auth_url = http://$CONTROLLER_ADMIN_NODE:5000/v2.0/
admin_user = glance
admin_password = service-password
admin_tenant_name = service
```

#### ■ /etc/glance/glance-registry.conf

```
[DEFAULT]
default_publisher_id = image.$CONTROLLER_ADMIN_NODE

[database]
connection = mysql://glance:service-password@$CONTROLLER_ADMIN_NODE/glance

[keystone_auth token]
auth_uri = http://$CONTROLLER_ADMIN_NODE:5000/v2.0
identity_uri = http://$CONTROLLER_ADMIN_NODE:35357
admin_user = glance
admin_password = service-password
admin_tenant_name = service
```

```
[oslo_messaging_rabbit]
rabbit_host=$CONTROLLER_ADMIN_NODE
```

- /etc/glance/glance-scrubber.conf

```
[DEFAULT]
auth_url = http://$CONTROLLER_ADMIN_NODE:5000/v2.0/
identity_uri = http://$CONTROLLER_ADMIN_NODE:35357
admin_user = glance
admin_password = service-password
admin_tenant_name = service

[database]
connection=mysql://glance:service-password@$CONTROLLER_ADMIN_NODE/glance
```

2. 启用 Glance SMF 服务。

```
controller# svcadm enable -rs glance-api glance-db glance-registry glance-scrubber
```

## 安装 Nova

本节介绍的是 Nova 端点服务配置，而非计算节点本身。

### ▼ 如何安装并配置 Nova

1. 通过在 /etc/nova/nova.conf 文件中取消注释或设置参数来配置 Nova。

```
[DEFAULT]
my_ip=$CONTROLLER_ADMIN_NODE_IP
host=$CONTROLLER_ADMIN_NODE
firewall_driver=nova.virt.firewall.NoopFirewallDriver

[database]
connection = mysql://nova:service-password@$CONTROLLER_ADMIN_NODE/nova

[glance]
host=$CONTROLLER_ADMIN_NODE

[keystone_authtoken]
auth_uri=http://$CONTROLLER_ADMIN_NODE:5000/v2.0/
identity_uri=http://$CONTROLLER_ADMIN_NODE:35357/
admin_user=nova
admin_password=service-password
admin_tenant_name=service

[neutron]
```

```
url=http://$CONTROLLER_ADMIN_NODE:9696
admin_username=neutron
admin_password=service-password
admin_tenant_name=service
admin_auth_url=http://$CONTROLLER_ADMIN_NODE:5000/v2.0

[oslo_messaging_rabbit]
rabbit_host=$CONTROLLER_ADMIN_NODE
```

2. 设置 `/etc/nova/api-paste.ini` 文件中的参数。

```
[filter:authtoken]
admin_user = nova
admin_password = service-password
admin_tenant_name = service
auth_uri = http://$CONTROLLER_ADMIN_NODE:5000/v2.0/
identity_uri = http://$CONTROLLER_ADMIN_NODE:35357
```

3. 启用 Nova SMF 服务。

```
controller# svcadm enable -rs nova-conductor
controller# svcadm enable -rs nova-api-osapi-compute \
nova-cert nova-scheduler
```

## 安装 Horizon

Horizon 用作 OpenStack 的 Web 门户。

### ▼ 如何配置 Horizon

1. 为 SSL/TLS 设置 Horizon 配置。

- a. 生成证书以供 Horizon 使用。

以下命令生成自签名的证书以供 Horizon 使用，并将 OpenStack 系统信息显示板配置文件复制到 Apache 配置文件目录中。有关创建自签名证书的更多信息，请参见《[SSL/TLS Strong Encryption: FAQ](#)》（《[SSL/TLS 强加密：常见问题解答](#)》）。

```
controller# export DASHBOARD=/etc/openstack_dashboard
controller# openssl req -new -x509 -nodes \
-out horizon.crt -keyout horizon.key
```

此时，按照提示提供信息，如国家/地区、省/市/自治区、城市、公司、组织、名称和电子邮件地址。然后继续移动键。

```
controller# mv horizon.crt horizon.key ${DASHBOARD}
controller# chmod 0644 ${DASHBOARD}/*
controller# chown websrvd:websrvd ${DASHBOARD}/*
```

```
controller# sed \  
-e "/SSLCertificateFile/s:/path.*:${DASHBOARD}/horizon.crt:" \  
-e "/SSLCACertificateFile/d" \  
-e "/SSLCertificateKeyFile/s:/path.*:${DASHBOARD}/horizon.key:" \  
< /etc/apache2/2.4/samples-conf.d/openstack-dashboard-tls.conf \  
> /etc/apache2/2.4/conf.d/openstack-dashboard-tls.conf
```

- b. 在 `/etc/apache2/2.4/conf.d/openstack-dashboard-tls.conf` 文件中，在以下参数上指定 Horizon 软件包的站点地址和服务器名称：

```
RedirectPermanent /horizon https://controller-fqdn/horizon  
ServerName controller-fqdn
```

2. 启动 Apache 服务。

```
controller# svcadm enable apache24
```

## 安装 Cinder

Cinder 配置必须至少指定以下信息：

- 要使用 Keystone 验证的授权信息。
- 要创建的卷类。

### ▼ 如何安装并配置 Cinder

此过程中的步骤引用 Cinder 端点服务（而不是 Cinder 或卷节点）的配置。

1. 通过在 `/etc/cinder/cinder.conf` 文件中取消注释或设置参数来配置 Cinder。  
对于 `volume_driver` 参数，有多个驱动程序可供选择。在以下示例中，仅显示了 `volume_driver` 的选定驱动程序。其他注释掉的可用驱动程序已排除在外。

```
[DEFAULT]  
volume_driver=cinder.volume.drivers.solaris.zfs.ZFSISCSIDriver  
my_ip=${CONTROLLER_ADMIN_NODE}  
  
[database]  
connection = mysql://cinder:service-password@${CONTROLLER_ADMIN_NODE}/cinder  
  
[keystone_authtoken]  
auth_uri = http://${CONTROLLER_ADMIN_NODE}:5000/v2.0  
identity_uri = http://${CONTROLLER_ADMIN_NODE}:35357  
admin_user = cinder
```

```
admin_password = service-password
admin_tenant_name = service

[oslo_messaging_rabbit]
rabbit_host=${CONTROLLER_ADMIN_NODE}
```

2. 配置 `/etc/cinder/api-paste.ini` 文件中的参数。

```
[filter:authtoken]
admin_tenant_name = service
admin_user = cinder
admin_password = service-password
```

3. 如果配置 iSCSI 目标，则应启用相应的 SMF 服务。

```
controller# svcadm enable iscsi/target stmf
```

4. 启用 Cinder SMF 服务。

```
controller# svcadm enable -rs cinder-db
controller# svcadm enable -rs cinder-api cinder-scheduler
```

另请参见 另请参见《[How to Build OpenStack Block Storage on ZFS](#)》（《如何在 ZFS 上生成 OpenStack 块存储》）。

## 安装 Neutron

在本章介绍的体系结构中，Neutron API 服务在控制器节点上运行。

### ▼ 如何安装并配置 Neutron

1. 通过在以下配置文件中取消注释或设置参数来配置 Neutron：

- `/etc/neutron/neutron.conf`

```
[DEFAULT]
host=${CONTROLLER_ADMIN_NODE}

[keystone_authtoken]
auth_uri = http://${CONTROLLER_ADMIN_NODE}:5000/v2.0
identity_uri = http://${CONTROLLER_ADMIN_NODE}:35357
admin_user = neutron
admin_password = service-password
admin_tenant_name = service
```

```
[database]
connection = mysql://neutron:service-password@$CONTROLLER_ADMIN_NODE/neutron
```

```
[oslo_messaging_rabbit]
rabbit_host=$CONTROLLER_ADMIN_NODE
```

- /etc/neutron/plugins/evs/evs\_plugin.ini

```
[EVS]
evs_controller = ssh://evsuser@$CONTROLLER_ADMIN_NODE
```

- /etc/neutron/dhcp\_agent.ini

```
[DEFAULT]
evs_controller = ssh://evsuser@$CONTROLLER_ADMIN_NODE
```

- /etc/neutron/l3\_agent.ini

```
evs_controller = ssh://evsuser@$CONTROLLER_ADMIN_NODE
```

---

注 - 缺省情况下，云中项目的内部网络相互隔离。一个项目的网络只能相互通信，而不能与其他项目的网络通信。要允许云中的所有网络相互连接，而不管它们属于哪个项目，请使用以下参数设置编辑 /etc/neutron/l3\_agent.ini 文件：

```
allow_forwarding_between_networks = true
```

---

## 2. 设置要使用的 SSH 密钥对。

- a. 为 `evsuser`、`neutron` 和 `root` 用户创建 SSH 密钥对。

```
controller# su - evsuser -c "ssh-keygen -N '' \
-f /var/user/evsuser/.ssh/id_rsa -t rsa"
controller# su - neutron -c "ssh-keygen -N '' -f /var/lib/neutron/.ssh/id_rsa -t rsa"
controller# ssh-keygen -N '' -f /root/.ssh/id_rsa -t rsa
```

- b. 在 `evsuser` 的 `authorized_keys` 文件中组合 `evsuser`、`neutron` 和 `root` 的 SSH 密钥。

```
controller# cat /var/user/evsuser/.ssh/id_rsa.pub \
/var/lib/neutron/.ssh/id_rsa.pub /root/.ssh/id_rsa.pub >> \
/var/user/evsuser/.ssh/authorized_keys
```

- c. 测试 SSH 连接以接受将存储在 `known_host` 文件中的指纹。

在配置在每个提示符处指定 Yes。

```
controller# su - evsuser -c "ssh evsuser@$CONTROLLER_ADMIN_NODE true"
controller# su - neutron -c "ssh evsuser@$CONTROLLER_ADMIN_NODE true"
```

```
controller# ssh evsuser@$CONTROLLER_ADMIN_NODE true
```

### 3. 配置弹性虚拟交换机 (elastic virtual switch, EVS)。

---

注 - 以下子步骤专门为基于 VLAN 的网络配置 EVS。

要配置基于 VXLAN 的网络，请转至 [Operating Systems Documentation](#) (操作系统文档) 中您 Oracle Solaris 版本相应库中的《在 *Oracle Solaris* 中管理网络虚拟化和网络资源》。在该书中，请特别参阅“用例：为租户配置弹性虚拟交换机”。

有关如何配置平面网络的示例，请参见 [https://blogs.oracle.com/openstack/entry/configuring\\_the\\_neutron\\_l3\\_agent](https://blogs.oracle.com/openstack/entry/configuring_the_neutron_l3_agent)。

---

#### a. 设置 EVS 属性以指定 EVS 控制器的位置。

```
controller# evsadm set-prop -p controller=ssh://evsuser@$CONTROLLER_ADMIN_NODE
```

#### b. 配置 EVS 控制器的 l2-type、vlan-range 和 uplink-port 属性。

```
controller# evsadm set-controlprop -p property=value
```

如图 1 中所示，通常有多个网络接口来服务不同的子网。在设置 uplink-port 属性时，您可以将 VLAN 拆分到为子网提供服务的多个网络端口中。

以下示例说明如何设置 EVS 属性，包括拆分 VLAN。（可选）使用最后的命令显示所有 EVS 属性。

---

注 - 在将 VLAN 分布到网络端口中之前，首先定义 VLAN 范围。否则，无法配置 uplink-port 属性。

---

```
controller# evsadm set-controlprop -p l2-type=vlan
controller# evsadm set-controlprop -p vlan-range=1,200-300
controller# evsadm set-controlprop -p uplink-port=net0,vlan-range=1
controller# evsadm set-controlprop -p uplink-port=net1,vlan-range=200-250
controller# evsadm set-controlprop -p uplink-port=net2,vlan-range=251-300

controller# evsadm show-controlprop -o all
```

### 4. 启用 IP 转发。

```
controller# ipadm set-prop -p forwarding=on ipv4
controller# ipadm set-prop -p forwarding=on ipv6
```

### 5. 启动 IP 过滤器服务。

```
controller# svcadm enable -rs ipfilter
```

### 6. 启用 Neutron 服务器服务。

```
controller# svcadm enable -rs neutron-server neutron-dhcp-agent
```

## 配置计算节点

在计算节点中安装 VM 实例，并安装 nova-compute 守护进程。VM 实例提供各种服务，如 Web 应用程序和分析。可根据您的云的需要配置相应数量的计算节点。

要配置计算节点，请使用以下命令在系统上安装 OpenStack 组件和服务：

```
compute# pkg install openstack
```

在软件包安装完成后，配置您要在节点上运行的服务。

---

注 - 为更好地管理 Oracle Solaris 11 中 ZFS 与应用程序之间的内存使用情况，请在该节点上设置 `usr_reserve_hint_pct` 参数，如以下示例所示：

```
# echo "set user_reserve_hint_pct=80" >>/etc/system.d/$(site):kernel-zones-reserve
# reboot
```

其中，`site` 是一个随机标识符，例如公司的名称。

还要在不同的 OpenStack 节点上设置此参数。

有关此参数的更多信息，请在 <https://support.oracle.com> 中登录到您的 MOS 帐户，然后查看文档 1663862.1 《*Memory Management Between ZFS and Applications in Oracle Solaris 11.2*》。

---

### ▼ 如何配置计算节点

1. 配置 NTP 客户机。  
请参见“配置 NTP 客户机” [30]。
2. 重新启动远程访问守护进程 (Remote Access Daemon, RAD)。  
Nova 使用 RAD 与 Oracle Solaris Zones 框架通信。

```
compute1# svcadm restart rad:local
```

3. 通过在 `/etc/nova/nova.conf` 文件中取消注释或设置以下参数来配置 Nova。

```
[DEFAULT]
```



```

my_ip=$COMPUTE_ADMIN_NODE_IP
host=$COMPUTE_ADMIN_NODE_X
firewall_driver=nova.virt.firewall.NoopFirewallDriver
keystone_ec2_url=http://$CONTROLLER_ADMIN_NODE:5000/v2.0/ec2tokens

[database]
connection = mysql://nova:service-password@$CONTROLLER_ADMIN_NODE/nova

[glance]
host=$CONTROLLER_ADMIN_NODE

[keystone_authtoken]
auth_uri=http://$CONTROLLER_ADMIN_NODE:5000/v2.0/
identity_uri=http://$CONTROLLER_ADMIN_NODE:35357/
admin_user=nova
admin_password=service-password
admin_tenant_name=service

[neutron]
url=http://$CONTROLLER_ADMIN_NODE:9696
admin_username=neutron
admin_password=service-password
admin_tenant_name=service
admin_auth_url=http://$CONTROLLER_ADMIN_NODE:5000/v2.0

[oslo_messaging_rabbit]
rabbit_host=$CONTROLLER_ADMIN_NODE

```

4. 设置 `/etc/nova/api-paste.ini` 文件中的参数。

```

[filter:authtoken]
admin_user = nova
admin_password = service-password
admin_tenant_name = service
auth_uri = http://$CONTROLLER_ADMIN_NODE:5000/v2.0/
identity_uri = http://$CONTROLLER_ADMIN_NODE:35357

```

5. 在计算节点上设置 EVS。

a. 确保已安装 EVS 软件包。

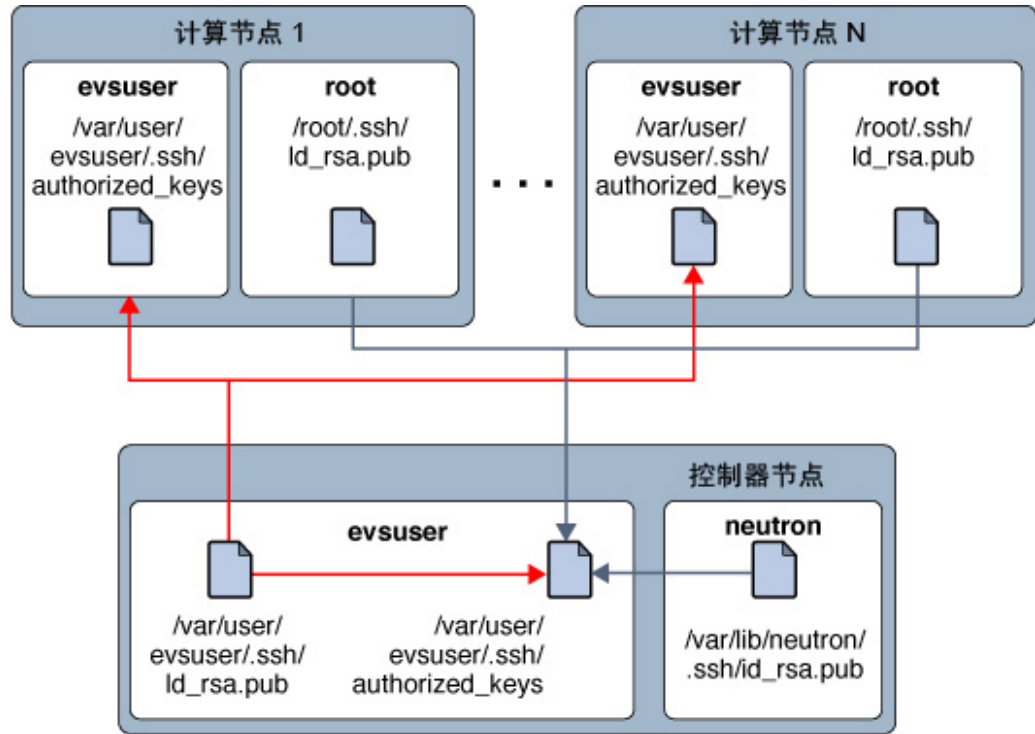
```
compute1# pkg info evs
```

b. 指定 EVS 控制器的位置。

```
compute1# evsadm set-prop -p controller=ssh://evsuser@$CONTROLLER_ADMIN_NODE
```

6. 配置控制器与计算节点之间的通信。

用于建立节点间通信的 SSH 密钥的分布类似下图：



- a. 在计算节点上为 root 用户创建一个 SSH 公钥。

```
compute1# ssh-keygen -N '' -f /root/.ssh/id_rsa -t rsa
```

- b. (可选) 检查 SSH 密钥的内容。

```
compute1# cat /root/.ssh/id_rsa.pub
```

- c. 将 SSH 密钥 `/root/.ssh/id_rsa.pub` 复制到控制器节点中的某个位置。

- d. 在该控制器节点上，将 SSH 密钥附加到 evsuser 的 `authorized_keys` 文件。

```
controller# cat location/id_rsa.pub >> /var/user/evsuser/.ssh/authorized_keys
```

- e. (可选) 验证是否已将计算节点中的 SSH 密钥附加到 `authorized_keys` 文件。

```
controller# cat /var/user/evsuser/.ssh/authorized_keys
```

输出应包含在计算节点上生成的 `/root/.ssh/id_rsa.pub` 的内容。

- f. 测试计算节点到控制器的 SSH 连接，并接受要存储在 `known_host` 文件中的指纹。在提示符处指定 `Yes` 进行确认。

```
compute1# ssh evsuser@$CONTROLLER_ADMIN_NODE true
```

7. 启用 Nova 计算服务。

```
compute1# svcadm enable nova-compute
```

## ▼ 如何启用控制台访问

使用此过程将能够基于用户的请求从浏览器使用 VM 实例的控制台。

1. 在每个计算节点上，根据具体情况执行下列步骤。
  - 如果可以从面向公众的网络访问计算节点的 IP 地址，请在 `/etc/nova/nova.conf` 文件的 `[DEFAULT]` 部分中设置以下参数。

```
[DEFAULT]
...
vnc_enabled = true
vncserver_listen = 0.0.0.0
novncproxy_port = 6080
novncproxy_base_url =http://FQDN:6080/vnc_auto.html
novncproxy_host = 0.0.0.0
...
```

其中，`FQDN` 表示计算节点的全限定域名或 IP 地址。

- 如果计算节点在专用网络中，请在 `/etc/nova/nova.conf` 文件的 `[DEFAULT]` 部分中设置以下参数。

```
[DEFAULT]
...
vnc_enabled = true
vncserver_listen = internal-IP
novncproxy_port=6080
novncproxy_base_url = http://public-IP:6080/vnc_auto.html
vncserver_proxycient_address = internal-IP
```

- `internal-IP` – 计算节点在内部网络上的 IP 地址。
- `public-IP`—控制器主机的公共 IP 地址。

2. 如果可以从面向公众的网络访问计算节点的 IP 地址，则执行以下子步骤。否则，请跳到下一步。

- a. 启用 `nova-novncproxy` 服务。

```
compute# svcadm enable nova-novncproxy
```

- b. 重新启动 `nova-compute` 服务。

```
compute# svcadm restart nova-compute
```

3. 在控制器节点上，根据具体情况执行下列步骤。

- 如果可以从面向公众的网络访问计算节点的 IP 地址，请启用 `nova-consoleauth` 服务。

```
controller# svcadm enable nova-consoleauth
```

- 如果计算节点在专用网络中，请执行下列步骤：

- a. 在 `/etc/nova/nova.conf` 文件的 [DEFAULT] 部分中设置以下参数。

```
novncproxy_base_url=http://public-IP:6080/vnc_auto.html
```

其中，*public-IP* 是控制器主机的公共 IP 地址。

- b. 如下所示启用 Nova 服务：

```
controller# svcadm enable nova-consoleauth  
controller# svcadm enable nova-novncproxy
```

## 配置存储节点

控制器节点是在 OpenStack 设置中进行事务处理的所有数据的系统信息库。

要配置计算节点，请使用以下命令在系统上安装 OpenStack 组件和服务：

```
storage# pkg install openstack
```

在软件包安装完成后，配置您要在节点上运行的服务。

注 - 为更好地管理 Oracle Solaris 11 中 ZFS 与应用程序之间的内存使用情况，请在该节点上设置 `usr_reserve_hint_pct` 参数，如以下示例所示：

```
# echo "set user_reserve_hint_pct=80" >>/etc/system.d/Site:kernel-zones-reserve
# reboot
```

其中，`site` 是一个随机标识符，例如公司的名称。

还要在不同的 OpenStack 节点上设置此参数。

有关此参数的更多信息，请在 <https://support.oracle.com> 中登录到您的 MOS 帐户，然后查看文档 1663862.1 《*Memory Management Between ZFS and Applications in Oracle Solaris 11.2*》。

## ▼ 如何配置块存储节点

此过程介绍块存储的典型配置。对于配置存储组件的其他选项，请参见第 6 章 [用于 Cinder 配置和部署的选项](#)。

1. 配置 NTP 客户机。  
请参见“[配置 NTP 客户机](#)” [30]。
2. 通过在 `/etc/cinder/cinder.conf` 文件中取消注释或设置参数来配置 Cinder。

```
[DEFAULT]
san_is_local=true
volume_driver=cinder.volume.drivers.solaris.zfs.ZFSISCSIDriver
my_ip=$VOLUME_IP
glance_host=$CONTROLLER_ADMIN_NODE
zfs_volume_base=cinder/cinder

[database]
connection = mysql://cinder:service-password@$CONTROLLER_ADMIN_NODE/cinder

[keystone_authtoken]
auth_uri = http://$CONTROLLER_ADMIN_NODE:5000/v2.0
identity_uri = http://$CONTROLLER_ADMIN_NODE:35357
admin_user = cinder
admin_password = service-password
admin_tenant_name = service

[oslo_messaging_rabbit]
rabbit_host=$CONTROLLER_ADMIN_NODE
```

3. 设置 `/etc/cinder/api-paste.ini` 文件中的参数。

```
[filter:authtoken]
```

```
admin_tenant_name = service
admin_user = cinder
admin_password = service-password
```

4. 启动 Cinder 服务。

```
storage# svcadm enable -rs cinder-db cinder-volume:default cinder-volume:setup
storage# svcadm enable -rs iscsi/target
```

## 配置 Swift 对象存储

Swift 是 OpenStack 对象存储项目。它提供云存储软件来使您能够通过简单的 API 存储和检索大型数据。该服务最适合用于存储可以无限增长的非结构化数据。

有关 Swift 的更多信息，请参阅 OpenStack 社区中《[OpenStack Cloud Administrator Guide](#)》（《OpenStack 云管理员指南》）中的 Object Storage（“对象存储”）一章。

该 OpenStack 社区文档建议在生产模式下至少为 Swift 部署使用 6 个节点。这些节点包括 1 个 Swift 代理控制器和 5 个 Swift 控制器节点。不过，为了与前面的配置过程所基于的 3 节点体系结构保持一致，本指南介绍的是一个 3 节点部署。您以后可以根据需要添加控制器节点。

### ▼ 如何配置 Swift 代理控制器服务节点

此任务假定您已在为 Swift 指定的节点上安装了 OpenStack 软件包。请在[“配置存储节点” \[44\]](#)中查看要使用的安装命令。

1. 安装 Swift 软件包。

```
proxy-node # pkg install swift swiftclient
```

2. 创建 ZFS 数据集。

```
proxy-node # /usr/sbin/zfs create -o mountpoint=none rpool/export/swift
proxy-node # /usr/sbin/zfs create -o mountpoint=/srv rpool/export/swift/srv
proxy-node # /usr/sbin/zfs create -p rpool/export/swift/srv/node/disk0
proxy-node # /usr/bin/chown -R swift:swift /srv
```

3. 执行以下八进制转储。

保留转储的值。这两个值在后续步骤中称为 \$OD\_1 和 \$OPD\_2。

```
proxy-node # od -t x8 -N 8 -A n < /dev/random
proxy-node # od -t x8 -N 8 -A n < /dev/random
```

4. 使用以下参数编辑 `/etc/swift/swift.conf` 文件。

```
[swift-hash]
swift_hash_path_suffix = $OD_1
swift_hash_path_prefix = $OD_2
```

5. 使用以下参数编辑 `/etc/swift/proxy-server.conf` 文件。

```
[DEFAULT]
bind_port = 8080

[filter:tempauth]
use = egg:swift#tempauth

operator_roles = admin, swiftoperator

[filter:authtoken]
auth_uri = http://$CONTROLLER_IP:5000/
identity_uri = http://$CONTROLLER_IP:35357
admin_user = swift
admin_password = swiftpass
admin_tenant_name = service

[filter:cache]
memcache_servers = $CONTROLLER_IP:11211

[oslo_messaging_rabbit]
rabbit_host=$CONTROLLER_ADMIN_NODE
```

6. 启用 `memcached` 守护进程。

```
proxy-node # svcadm enable -rs memcached
```

7. 构建环。

```
proxy-node # cd /etc/swift
proxy-node # swift-ring-builder account.builder create 18 3 1
proxy-node # swift-ring-builder container.builder create 18 3 1
proxy-node # swift-ring-builder object.builder create 18 3 1
proxy-node # swift-ring-builder account.builder add r1z1-$STORAGE_IP_1:6002/disk0 100
proxy-node # swift-ring-builder container.builder add r1z1-$STORAGE_IP_1:6001/disk0
100
proxy-node # swift-ring-builder object.builder add r1z1-$STORAGE_IP_1:6000/disk0 100
proxy-node # swift-ring-builder account.builder add r1z1-$STORAGE_IP_2:6002/disk0 100
proxy-node # swift-ring-builder container.builder add r1z1-$STORAGE_IP_2:6001/disk0
100
proxy-node # swift-ring-builder object.builder add r1z1-$STORAGE_IP_2:6000/disk0 100
proxy-node # swift-ring-builder account.builder rebalance
proxy-node # swift-ring-builder container.builder rebalance
proxy-node # swift-ring-builder object.builder rebalance
proxy-node # >chown -R swift:swift /etc/swift
```

8. 启用 Swift 服务。

```
proxy-node # svcadm enable swift-proxy-server
```

## ▼ 如何配置对象存储节点

在您要设置的每个对象控制器节点上重复此过程。

1. 安装 Swift 软件包。

```
storage-node # pkg install swift swiftclient
```

2. 创建 ZFS 数据集。

```
storage-node # /usr/sbin/zfs create -o mountpoint=none rpool/export/swift
storage-node # /usr/sbin/zfs create -o mountpoint=/srv rpool/export/swift/srv
storage-node # /usr/sbin/zfs create -p rpool/export/swift/srv/node/disk0
storage-node # /usr/bin/chown -R swift:swift /srv
```

3. 如下所述从代理服务器节点复制文件：

- a. 将 `/etc/swift/swift.conf` 文件从代理服务器节点复制到当前节点的 `/etc/swift` 目录。
- b. 将以下文件从代理服务器节点复制到当前节点的 `/etc/swift` 目录。
  - `account.ring.gz`
  - `container.ring.gz`
  - `object.ring.gz`

4. 启用 Swift 复制器服务。

```
storage-node # svcadm enable swift-replicator-rsync
```

5. 设置当前节点的 `/etc/swift` 目录的所有权。

```
storage-node # chown -R swift:swift /etc/swift
```

6. 启用所有 Swift 服务。

```
storage-node # for x in `svcs -a -o SVC | fgrep swift | \
    egrep "account|container|object" | sort` \
    do \
        echo Starting $x \
        svcadm enable $x \
    done
```

7. 在控制器节点上，使用户能够访问和运行 Swift 服务。



- a. 为 Swift 设置全局 shell 变量。

```
controller# export OS_USERNAME=swift
controller# export OS_PASSWORD=service-password
controller# export OS_PROJECT_NAME=service
controller# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0
```

- b. 添加 Keystone swiftoperator 角色。

```
controller# openstack role create --name swiftoperator
```

- c. 将 swiftoperator 角色分配给 Swift 服务的授权用户。

```
controller# openstack user role add --user user-name \
--role swiftoperator --project tenant-ID
```

接下来的步骤 要使用户可以使用云，请完成[第 4 章 安装后和配置任务](#)中所述的准备工作。



## 安装后和配置任务

---

本章介绍完成初始 OpenStack 安装和配置的过程。本章涵盖以下主题：

- “为 OpenStack 项目准备外部网络” [51]
- “为 Glance 系统信息库准备映像” [59]

### 为 OpenStack 项目准备外部网络

外部网络提供了云中专用网络和公共网络的连接。

#### 关于提供商路由器

路由器提供项目 VM 实例与更广泛网络的连接。此路由器由所有项目网络共享。因为只有一个路由器，所以项目网络无法使用重叠的 IP 地址。

该路由器在连接路由器与外部网络的接口上执行双向网络地址转换 (network address translation, NAT)。项目可以有許多浮动 IP (公共 IP)，具体数量取决于项目需要或浮动 IP 配额所允许的数量。这些浮动 IP 与需要外部连接的 VM 实例关联。

创建路由器需要配置 Neutron L3 代理。此代理将自动在分配给 Nova 实例的地址和浮动 IP 地址之间创建一对一的 NAT 映射。L3 代理还支持专用网络之间的通信。

缺省情况下，会禁用同一个项目的专用网络之间的路由。要更改此行为，请将 `/etc/neutron/l3_agent.ini` 配置文件中的 `allow_forwarding_between_networks` 设置为 `True`。设置该参数后，重新启动 `neutron-l3-agent` SMF 服务。

#### ▼ 如何针对外部网络配置路由器

此过程介绍了如何针对外部网络创建路由器。部分步骤需要编辑配置文件。因此，在此过程中使用终端窗口比使用 Horizon 系统信息显示板更方便。

在安装了 Neutron 服务的节点上执行以下步骤。本文档根据前面章节所述样例体系结构定位控制器节点上的服务。

开始之前 确保如[如何安装并配置 Neutron \[37\]](#)中所述完成了 Neutron 的配置。

1. 如果 IP 过滤器服务已禁用，请启动它。

```
controller# svcadm enable -rs ipfilter
```

2. 如果在主机上禁用了 IP 转发，请将其启用。

```
controller# ipadm set-prop -p forwarding=on ipv4
```

3. 为 Neutron 设置全局 shell 变量。

```
controller# export OS_USERNAME=neutron
controller# export OS_PASSWORD=service-password
controller# export OS_PROJECT_NAME=service
controller# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0
```

4. 创建提供商路由器。

```
controller# neutron router-create router-name
```

该命令显示路由器名称及对应的 ID。使用此 ID 在下一步中更新配置文件。

5. 确保 `/etc/neutron/l3_agent.ini` 文件包含设定的以下参数。

```
router_id=routerID    从上一步获取的 ID。
```

6. 启用 `neutron-l3-agent` SMF 服务。

```
controller# svcadm enable neutron-l3-agent
```

7. (可选) 显示关于路由器的信息。

在将外部网络添加到路由器之后，可以添加关于路由器的更多信息。

```
controller# neutron router-show router-name
```

#### 例 1 创建路由器

此示例说明如何针对外部网络创建路由器。

```
controller# svcadm enable -rs ipfilter
```

```
controller# ipadm set-prop -p forwarding=on ipv4
controller# ipadm set-prop -p forwarding=on ipv6
```

```

controller# export OS_USERNAME=neutron
controller# export OS_PASSWORD=service-password
controller# export OS_PROJECT_NAME=service
controller# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0

```

```

controller# neutron router-create ext-router
Created a new router:

```

```

+-----+-----+
| Field                | Value                                |
+-----+-----+
| admin_state_up      | True                                 |
| external_gateway_info |                                       |
| id                  | f89b24ed-42dd-48b0-8f4b-fd41887a3370 |
| name                | ext-router                           |
| status              | ACTIVE                               |
| project_id          | 7d1caf0854b24becb28df5c5cabf72cc    |
+-----+-----+

```

此时，更新 `/etc/neutron/l3_agent.ini` 文件中的 `router_id`。

```
router_id = f89b24ed-42dd-48b0-8f4b-fd41887a3370
```

然后，启用 L3 代理服务。

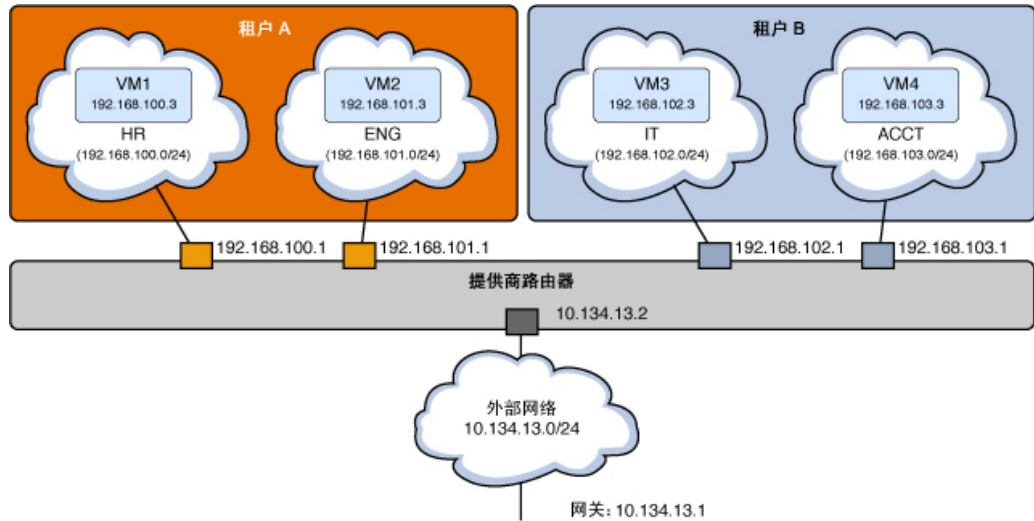
```
controller# svcadm enable neutron-l3-agent
```

## 创建外部网络

创建路由器后，接下来配置外部网络。通过外部网络的网关，云的内部网络中的用户可以连接到 Internet。

下图显示了单个路由器如何支持云项目的网络通信。

图 3 使用专用网络模型的提供商路由器



该图显示了以下内容：

- 两个项目 (租户 A 和 租户 B)
- 四个 VM (VM1、VM2、VM3 和 VM4)
- 四个子网 (HR、ENG、IT 和 ACCT)
- 路由器
- 外部网络

向内部网络提供对外部网络的访问时，VM 的 IP 地址将映射到分配给外部网络的浮动地址中的一个。

## ▼ 如何创建外部网络

此过程介绍了如何创建代表外部网络的虚拟网络。此虚拟网络不使用 DHCP。而是，创建浮动 IP 地址，并将其分配给项目，供这些项目下的 Nova VM 实例使用。这些步骤创建 VLAN 类型的网络，但是此过程适用于创建其他网络类型，例如平面网络。

外部网络的创建可以独立于内部网络的创建。

开始之前 完成弹性虚拟交换机的配置。有关更多信息，请参见[如何安装并配置 Neutron \[37\]](#)，尤其注意配置 EVS 的步骤。

1. 为 Neutron 设置全局 shell 变量。

```

controller# export OS_USERNAME=neutron
controller# export OS_PASSWORD=service-password
controller# export OS_PROJECT_NAME=service
controller# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0

```

2. (可选) 获取 VLAN 范围。

```

controller# evsadm show-controlprop -p vlan-range

```

3. 创建外部网络。

```

controller# neutron net-create --provider:network_type=vlan \
--provider:segmentation_id=VLAN-nbr \
--router:external network-name

```

此步骤假定根据[如何安装并配置 Neutron \[37\]](#)创建 VLAN 网络。segmentation\_id 是在配置 EVS 时定义其范围的 VLAN 网络的 VLAN ID。

---

注 - 如果要创建平面网络，则无需指定段 ID。

---

4. 创建外部网络的子网。

分配池由分配给子网的某个范围的浮动 IP 地址组成。

```

controller# neutron subnet-create --name subnet-name --disable-dhcp \
--allocation-pool start=start-IP,end=end-IP \
network-name subnet-IP

```

5. 向路由器添加外部网络。

```

controller# neutron router-gateway-set router-name ext-network-ID

```

---

注 - 缺省情况下，发出此命令时启用 SNAT。启用 SNAT 后，专用网络中的 VM 可以访问外部网络。不过，不能从云外部访问实例本身。要禁用 SNAT，请在 neutron router-gateway-set 子命令中指定 --disable-snat 选项。

---

6. (可选) 显示关于路由器的信息。

```

controller# neutron router-show router-name

```

## 例 2 创建外部网络

此示例说明如何创建外部网络，并将其准备就绪，以供云中的内部网络使用。

要创建平面网络，另请参见 <https://blogs.oracle.com/openstack/tags/juno> 的第 2 部分中提供的示例。

```

controller# export OS_USERNAME=neutron
controller# export OS_PASSWORD=service-password

```

```

controller# export OS_PROJECT_NAME=service
controller# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0

controller# evsadm show-controlprop -p vlan-range
PROPERTY          PERM VALUE          DEFAULT  HOST
vlan-range        rw  1,200-300        --      --

controller# neutron net-create --router:external \
--provider:network_type=vlan --provider:segmentation_id=1 ext_network
Created a new network:
+-----+-----+
| Field          | Value          |
+-----+-----+
| admin_state_up | True           |
| id             | 08cf49c8-f28f-49c1-933d-bdb1017e0294 |
| name          | ext_network    |
| provider:network_type | vlan          |
| provider:segmentation_id | 1            |
| router:external | True           |
| shared        | False          |
| status        | ACTIVE         |
| subnets      |                |
| project_id    | 7d1caf0854b24becb28df5c5cabf72cc |
+-----+-----+

controller# neutron subnet-create --name ext_subnet --disable-dhcp \
--allocation-pool start=10.134.10.8,end=10.134.10.254 \
ext_network 10.134.10.0/24
Created a new subnet:
+-----+-----+
| Field          | Value          |
+-----+-----+
| allocation_pools | {"start": "10.134.10.8", "end": "10.134.10.254"} |
| cidr           | 10.134.10.0/24 |
| dns_nameservers |                |
| enable_dhcp    | False          |
| gateway_ip     | 10.134.13.1    |
| host_routes    |                |
| id            | fce503ff-f483-4024-b122-f2524e3edae1 |
| ip_version     | 4              |
| ipv6_address_mode |                |
| ipv6_ra_mode   |                |
| name          | ext_subnet     |
| network_id     | 08cf49c8-f28f-49c1-933d-bdb1017e0294 |
| project_id    | 7d1caf0854b24becb28df5c5cabf72cc |
+-----+-----+

controller# neutron router-gateway-set ext-router 08cf49c8-f28f-49c1-933d-bdb1017e0294
Set gateway for router ext-router

controller# neutron router-show ext-router
+-----+-----+
| Field          | Value          |
+-----+-----+

```



```

| admin_state_up      | True |
| external_gateway_info | {"network_id": "08cf49c8-f28f-49c1-933d-bdb1017e0294", |
|                     | "enable_snat": true, |
|                     | "external_fixed_ips": |
|                     | [{"subnet_id": "fce503ff-f483-4024-b122-f2524e3edae1", |
|                     | "ip_address": "10.134.10.8"}]} |
| id                  | f89b24ed-42dd-48b0-8f4b-fd41887a3370 |
| name                 | ext-router |
| status               | ACTIVE |
| project_id           | 7d1caf0854b24becb28df5c5cabf72cc |
+-----+-----+

```

- 另请参见
- [如何观察 L3 代理配置 \[58\]](#)。
  - [“已知限制” \[126\]](#)。

## ▼ 如何向内部网络提供外部连接

使用此过程以允许内部网络访问更广泛的公共网络。此过程假定特定项目已存在内部网络。要使用系统信息显示板创建项目内部网络，请参见[“创建项目的内部网络” \[70\]](#)。

开始之前 在继续操作之前，请获取需要公共访问权限的子网名称。

1. 为 Neutron 设置全局 shell 变量。

```

controller# export OS_USERNAME=neutron
controller# export OS_PASSWORD=service-password
controller# export OS_PROJECT_NAME=service
controller# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0

```

2. 标识需要外部访问权限的子网的 ID。

```

controller# neutron subnet-list | grep subnet-name

```

3. (可选) 获取路由器的名称。

```

controller# neutron router-list

```

4. 添加子网的 ID 作为路由器接口。

```

controller# neutron router-interface-add router-name subnetID

```

例 3 将内部网络连接到外部网络

此示例使用了在[如何为项目配置网络 \[70\]](#)中创建的 HR 内部网络。HR 网络（其子网为 HR\_Subnet）需要公共网络访问权限。

```

controller# neutron subnet-list | grep HR_Subnet
| b6feff42-36aa-4235- | HR_Subnet | 10.132.30.0/24 | {"start": "10.132.30.2", |

```

```
| 9fe0-ac5de6b43af3 | | "end": "10.132.30.254"} |
```

```
controller# neutron router-list
```

```
+-----+-----+-----+
| id          | name      | external_gateway_info  |
+-----+-----+-----+
| f89b24ed-42dd-48b0-8f4b-fd41887a3370 | ext-router | {"network_id": "6c4c1823-a203-43b1-9674-ddb5ff4185fc", "enable_snat": true, "external_fixed_ips": [{"subnet_id": "83d9b40f-cc61-4696-b22e-b4cbc2aa3872", "ip_address": "10.132.10.8"}]} |
+-----+-----+-----+
```

```
controller# neutron router-interface-add ext-router b6feff42-36aa-4235-9fe0-ac5de6b43af3
Added interface b6feff42-36aa-4235-9fe0-ac5de6b43af3 to router ext-router.
```

## ▼ 如何观察 L3 代理配置

您可以使用诸如 ipf、ippool 和 ipnat 等 IP 过滤器命令以及诸如 dladm 和 ipadm 等网络命令观察 neturon-l3-agent 执行的配置并对其进行故障排除。

### 1. 显示 neutron-l3-agent 创建的 VNIC。

```
network# dladm show-vnic
LINK          OVER          SPEED  MACADDRESS    MACADDRTYPE  VIDS
l3i7843841e_0_0 net1          1000   2:8:20:42:ed:22 fixed         200
l3i89289b8e_0_0 net1          1000   2:8:20:7d:87:12 fixed         201
l3ed527f842_0_0 net0          100    2:8:20:9:98:3e fixed          0
```

### 2. 显示 neutron-l3-agent 创建的 IP 地址。

```
network# ipadm
NAME          CLASS/TYPE  STATE  UNDER  ADDR
l3ed527f842_0_0 ip          ok     --     --
  l3ed527f842_0_0/v4 static     ok     --     10.134.10.8/24
  l3ed527f842_0_0/v4a static     ok     --     10.134.10.9/32
l3i7843841e_0_0 ip          ok     --     --
  l3i7843841e_0_0/v4 static     ok     --     192.168.100.1/24
l3i89289b8e_0_0 ip          ok     --     --
  l3i89289b8e_0_0/v4 static     ok     --     192.168.101.1/24
```

### 3. 显示 IP 过滤器规则。

```
network# ipfstat -io
empty list for ipfilter(out)
block in quick on l3i7843841e_0_0 from 192.168.100.0/24 to pool/4386082
block in quick on l3i89289b8e_0_0 from 192.168.101.0/24 to pool/8226578
network# ippool -l
table role = ipf type = tree number = 8226578
```

```
{ 192.168.100.0/24; };
table role = ipf type = tree number = 4386082
{ 192.168.101.0/24; };
```

#### 4. 显示 IP NAT 规则。

```
network# ipnat -l
List of active MAP/Redirect filters:
bimap l3ed527f842_0_0 192.168.101.3/32 -> 10.134.13.9/32
List of active sessions:
BIMAP 192.168.101.3 22 <- -> 10.134.13.9 22 [10.132.146.13 36405]
```

## 为 Glance 系统信息库准备映像

映像是云中 VM 实例的基础。映像是包含安装有可引导操作系统的虚拟磁盘的单个文件。映像提供用于创建一个或多个 VM 的模板。因此，要在云中置备 VM，必须首先创建映像。

Glance 即 OpenStack 映像服务，针对磁盘和服务器映像提供存储、搜索、注册以及交付服务。注册表服务器是向客户机提供映像元数据信息的映像服务。映像高速缓存由映像服务用于在本地主机上获取映像，而非每次请求映像时从映像服务器重新下载映像。

可以将多个映像上载到 Glance 系统信息库。最佳做法是上载要在云上部署的不同系统类型的映像。例如，创建非全局区域、内核区域以及全局区域的已归档映像。然后，可以选择相应的模板来快速部署其中一个类型的 VM。

## 创建映像

要在 Oracle Solaris 中创建 OpenStack 映像，可以使用统一归档文件功能。使用 `archiveadm` 命令，可以从全局、非全局和内核区域中创建新的统一归档文件 (Unified Archive, UA)。

UA 可以是克隆归档文件，也可以是恢复归档文件。克隆归档文件基于当前活动的引导环境。此归档文件不包括 OS 实例（如不活动的引导环境）中的任何系统配置信息。而是安装程序强制进行重新配置，或者使用您在系统配置 (system configuration, SC) 的配置文件中提供的配置信息。恢复归档文件包括所有引导环境和系统配置信息。因此，如果要将系统的所有信息包括在一个 UA 中，请创建一个恢复归档文件。有关 UA 的更多信息，请参见《使用统一归档文件进行系统恢复和克隆》。此书位于 [Operating Systems Documentation](#) (操作系统文档) 中您 Oracle Solaris 版本相应的库中。

以后，如果配置完全可操作，还可以通过创建现有 VM 实例的快照来创建映像。对于这种情况，VM 实例已在云中。因此，要使用的命令是 `nova image-create` 而非 `archiveadm`。`nova` 命令生成正在运行的 VM 实例的快照来创建映像。

还可以使用定制映像进行数据备份或者挽救 VM 实例。挽救映像是当 VM 实例处于 rescue 模式时引导的特殊映像类型。挽救映像允许管理员挂载 VM 实例的文件系统以更正问题。

在 Oracle Solaris 中，创建 OpenStack 映像分为以下三个阶段：

1. 创建区域。
2. 创建区域的 UA。
3. 将 UA 上载到 Glance。

这些阶段可合并成以下过程。

## ▼ 如何创建 OpenStack 的映像

除了创建区域的命令语法外，所有剩下的步骤可用于创建和上载非全局区域和内核区域的映像。

在此过程中，创建区域的步骤仅提供基本的命令。有关创建区域的完整说明，请参见关于安装、关闭、停止、卸载和克隆非全局区域的《创建和使用 *Oracle Solaris* 区域》。此书位于 [Operating Systems Documentation](#)（操作系统文档）中您 Oracle Solaris 版本相应的库中。

1. 在任何系统上，创建区域，然后登录到该区域。  
登录后，按照提示提供信息。

```
global# zonecfg -z zone-name create
global# zoneadm -z zone-name install
global# zoneadm -z zone-name boot
global# zlogin -C zone-name
```

---

注 - 此步骤可能需要一些时间才能完成。

---

2. 为 OpenStack root 登录访问启用 root SSH。

```
global# zlogin zone-name
root@zone-name# sed /^PermitRootLogin/s/no$/without-password/ < /etc/ssh/sshd_config
> /system/volatile/sed.$$
root@zone-name# cp /etc/ssh/sshd_config /etc/ssh/sshd_config.orig
root@zone-name# cp /system/volatile/sed.$$ /etc/ssh/sshd_config
root@zone-name# exit
```

3. 为区域创建 UA。

```
global# archiveadm create -z zone-name /var/tmp/archive-name.uar
```

4. 将 UA 传输到安装了 Glance 的系统。

本文档假定 Glance 位于控制器节点上。

## 5. 为 Glance 设置全局 shell 变量。

```
controller# export OS_USERNAME=glance
controller# export OS_PASSWORD=service-password
controller# export OS_PROJECT_NAME=service
controller# export OS_AUTH_URL=$CONTROLLER_ADMIN_NODE:5000/v2.0
```

## 6. 将 UA 上传到 Glance 系统信息库。

```
controller# glance --os-image-api-version 2 image-create \
  --container-format bare --disk-format raw \
  --visibility visibility-mode --name "image-name" \
  --property architecture=system-arch \
  --property hypervisor_type=solariszones \
  --property vm_mode=solariszones --file path-to-archive-file
```

*system-arch*            系统的体系结构，可以是 x86\_64 或 sparc64。

*visibility-mode*        映像辅助功能的范围，可以是 public 或 private。

## 显示有关映像的信息

要显示映像信息，可以使用 nova 命令或 glance 命令。

```
$ nova image-list
+-----+-----+-----+-----+
| ID                | Name                | Status | Server |
+-----+-----+-----+-----+
| 4dfbfd4f-2de5-4251-832c-e35a4a4145ee | Solaris Non-global Zone | ACTIVE |        |
+-----+-----+-----+-----+
```

glance image-list 命令显示其他信息，例如，不同映像的磁盘格式、容器格式以及大小。

nova image-show 和 glance image-show 命令显示有关特定映像的信息。每个命令生成有关映像的不同输出。

```
$ nova image-show 'Solaris Non-global Zone'
+-----+-----+-----+-----+
| Property          | Value              |
+-----+-----+-----+-----+
| OS-EXT-IMG-SIZE:size | 845025280          |
| created           | 2015-11-19T14:46:38Z |
| id                | 4dfbfd4f-2de5-4251-832c-e35a4a4145ee |
| metadata architecture | x86_64            |
| metadata hypervisor_type | solariszones      |
| metadata vm_mode    | solariszones      |
+-----+-----+-----+-----+
```

```
| minDisk          | 0 |
| minRam           | 0 |
| name             | Solaris Non-global Zone |
| progress         | 100 |
| status           | ACTIVE |
| updated          | 2015-11-19T14:46:42Z |
+-----+-----+

$ glance image-show 'Solaris Non-global Zone'
+-----+-----+
| Property          | Value |
+-----+-----+
| Property 'architecture' | x86_64 |
| Property 'hypervisor_type' | solariszones |
| Property 'vm_mode' | solariszones |
| checksum          | ba9b9eeddb467833d725c8750a46e004 |
| container_format  | bare |
| created_at        | 2015-11-19T14:46:38 |
| deleted           | False |
| disk_format       | raw |
| id                | 4dfbfd4f-2de5-4251-832c-e35a4a4145ee |
| is_public         | True |
| min_disk          | 0 |
| min_ram           | 0 |
| name              | Solaris Non-global Zone |
| owner             | 7d1caf0854b24becb28df5c5cabf72cc |
| protected         | False |
| size              | 845025280 |
| status            | active |
| updated_at        | 2015-11-19T14:46:42 |
+-----+-----+
```

---

注 - 可以从 Horizon 系统信息显示板获取同样的映像信息。

---

## 使用 Glance 映像创建脚本

glance image-create 命令可以上载映像并同时设置所有属性值。下面的脚本显示如何确保在 architecture 属性设置为当前主机的体系结构的情况下上载映像：

```
#!/bin/ksh

# Upload Unified Archive image to glance with proper Solaris decorations

arch=$(archiveadm info -p $1|grep ^archive|cut -d '|' -f 4)

if [[ "$arch" == "i386" ]]; then
    imgarch=x86_64
else
    imgarch=sparc64
fi
```

```
name=$(basename $1 .uar)

export OS_USERNAME=glance
export OS_PASSWORD=glance
export OS_TENANT_NAME=service
export OS_AUTH_URL=http://controller-name:5000/v2.0

glance image-create --name $name --container-format bare --disk-format raw --owner service
--file $1 --is-public True --property architecture=$imgarch --property
hypervisor_type=solariszones
--property vm_mode=solariszones --progress
```





## 使用云

---

本章介绍如何在云上执行不同的管理任务。可以使用系统信息显示板或命令行执行这些任务。在系统信息显示板上，“Project”（项目）选项卡下的任务仅需要成员角色，而“Admin”（管理员）选项卡下的任务则需要管理特权。要在单个登录会话中在系统信息显示板上执行某个项目的任务，应同时具有该项目的成员角色和管理员角色。

本章涵盖以下主题：

- “创建项目和用户” [68]
- “创建项目的内部网络” [70]
- “创建和引导 VM 实例” [73]
- “管理风格” [78]
- “管理 VM 实例” [82]

## 访问 OpenStack 系统信息显示板

完成 OpenStack 的安装及安装后配置任务之后，请登录 OpenStack 系统信息显示板，以查看云中的可用资源。

### ▼ 如何访问 OpenStack 系统信息显示板

1. 登录可以连接到 OpenStack 系统的任意系统。
2. 配置您的浏览器。
  - a. 启用 JavaScript。
  - b. 保存 Cookie。
3. 在浏览器的位置或地址栏中，输入以下位置：

`http://system/horizon/`

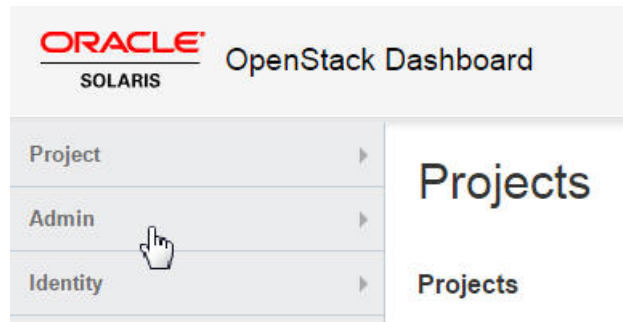
`system` 是在其中安装 OpenStack 统一归档文件并且在 Apache Web 服务器之下运行 Horizon OpenStack 服务的 OpenStack 系统的名称或 IP 地址。

如果在内核区域安装统一归档文件，则 OpenStack 系统为内核区域，`system` 是内核区域的名称或 IP 地址。

4. 在登录屏幕上提供以下信息：
  - 用户名：`admin`
  - 口令：`secrete`

## 浏览系统信息显示板

您最初以 `admin` 用户身份登录 Horizon 系统信息显示板，这将打开项目 `demo` 的登录页。因为 `admin` 对 `demo` 有管理特权，因此该页面会在左侧面板上显示 3 个选项卡：“Project”（项目）、“Admin”（管理员）和 “Identity”（身份）。没有管理特权的用户只能看到 “Project”（项目）和 “Identity”（身份）选项卡。



“Admin”（管理员）面板中的 “Usage Summary Overview”（使用情况摘要概况）页是缺省的云管理员视图。

图 4 OpenStack 系统信息显示板的 "Admin" (管理员) 面板的 "Overview" (概况) 窗口



"Admin" (管理员) 面板上的选项提供以下功能：

- 云中正在使用的 Nova 实例和 Cinder 卷的总体视图
- 查看和编辑定义 VM 实例特征的风格定义的功能，例如以下特征：
  - 虚拟 CPU 的数目
  - 内存量
  - 分配的磁盘空间
  - 底层 Oracle Solaris 区域的标记：solaris 表示非全局区域，solaris-kz 表示内核区域
- 创建虚拟网络和路由器以供云管理员使用的功能
- 查看和编辑项目的功能，该功能组合并分离虚拟计算资源的所有权
- 查看和编辑用户的功能，用户是使用云资源的人员或服务

单节点配置的 OpenStack UA 提供了以下预配置的资源：

- 两个映像：Solaris 非全局区域和 Solaris 内核区域
- 两个项目或租户：demo 和 service  
demo 项目是缺省项目，只有 admin 一个成员。  
云管理员使用 service 项目创建将在多个项目间共享的资源。例如，在文档中，在 service 项目中创建 Neutron 路由器，从而由所有项目共享该路由器。在 OpenStack 设置中，不要将 service 项目用于任何其他用途。OpenStack 服务通过特定于服务的用户方式来相互通信，这些用户在 service 项目中都拥有管理特权。
- 十种风格

要查看包括在 OpenStack 统一归档文件中的系统信息库中预配置的映像，请单击以下选项卡之一：

- "Admin" (管理员) -> "System" (系统) -> "Images" (镜像) 选项卡。
- "Project" (项目) -> "Compute" (计算) -> "Images" (镜像) 选项卡。

要列出自动提供的 Oracle Solaris 风格，请单击 "Admin" (管理员) -> "System" (系统) -> "Flavors" (云主机类型) 选项卡。

以下视频演示提供了系统信息显示板的完整概述：

- [The OpenStack Dashboard - Part 1](#) (OpenStack 系统信息显示板 – 第 1 部分)
- [The OpenStack Dashboard - Part 2](#) (OpenStack 系统信息显示板 – 第 2 部分)

有关可在系统信息显示板上执行的的任务的信息，请参见[第 5 章 使用云](#)。

## 创建项目和用户

以 admin 用户身份进行首次 OpenStack 登录时，将转到 demo 项目的登录页。从此项目开始，可以继续创建其他项目。

### ▼ 如何创建项目及分配用户

使用此过程可创建新项目或租户并用新用户填充它们。

1. 在浏览器中打开类似如下链接的 URL，然后以云管理员身份登录：

`http://system/horizon/`

此时将显示 demo 项目登录页。

有关系统信息显示板的简要概述，请参见[“浏览系统信息显示板” \[66\]](#)。

2. 在左侧面板上，选择 "Identity" (身份) -> "Projects" (项目) 选项卡。  
此时将显示缺省的 demo 和 service。
3. 单击 "Create Project" (创建项目)。
4. 在 "Project Information" (项目信息) 选项卡上，指定项目的名称以及项目说明。  
在创建了新项目后，会将其添加到项目列表中。
5. 选择 "Identity" (身份) -> "Users" (用户) 选项卡。  
此时将显示 demo 和 service 的缺省用户。
6. 单击 "Create User" (创建用户)。
7. 提供有关相应字段的必需信息。

- a. 指定新用户名称和分配的口令。
- b. 从 "Primary Project" (主项目) 下拉式菜单中选择新用户所属的项目。
- c. (可选) 从 "Role" (角色) 下拉式菜单中选择项目用户的角色。  
缺省情况下，项目的新用户具有成员角色。

## ▼ 如何使用现有用户填充项目

使用此过程将现有用户添加到新创建的项目。

1. 在缺省屏幕的左侧面板上，选择 "Identity" (身份) -> "Projects" (项目) 选项卡。
2. 对于要向其添加现有用户的项目，单击 "Modify Users" (修改用户)。  
此时将显示 "Edit Project" (编辑项目) 对话框。
3. 在 "All Users" (全部用户) 列表下，单击要添加到项目中的用户名右边的加号 (+)。  
缺省情况下，添加的用户在该项目上具有成员角色。

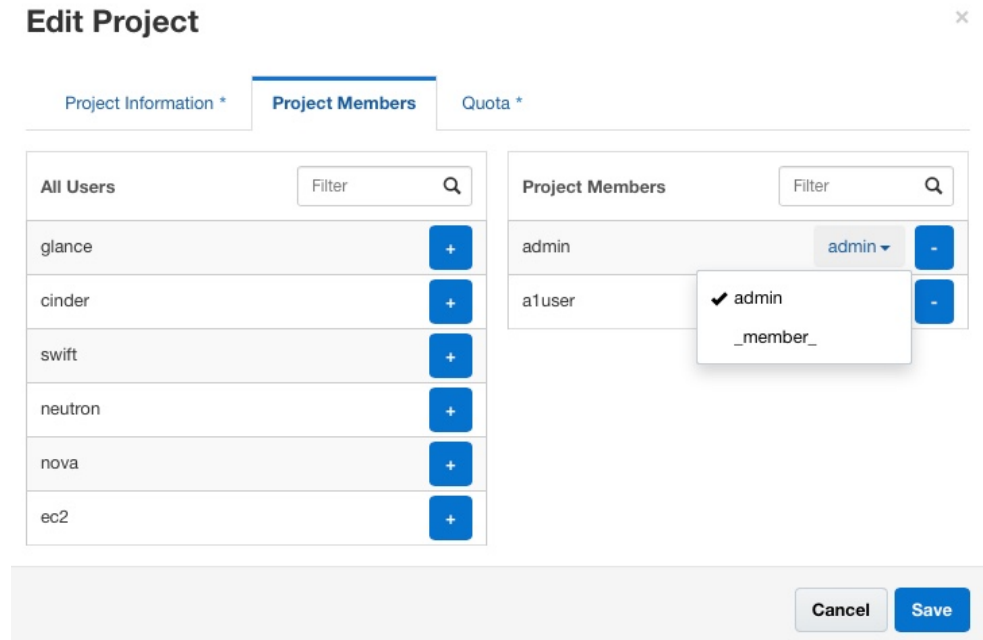
---

注 - 请不要将 service 项目的以下用户添加到其他项目：

- glance
- cinder
- swift
- neutron
- nova
- ec2

- 
4. (可选) 要修改项目的任何用户的角色，请执行以下步骤：
    - a. 从项目成员列表中，打开要更改其角色的用户的下拉式菜单。
    - b. 选择要分配给用户的新角色。

在此样例图中，当前项目的 "Project Members" (项目成员) 为 a1user 和 admin。admin 用户被授予了项目的管理特权。可以为用户同时分配 member 和 admin 角色。



## 创建项目的内部网络

一个项目可以有多个内部网络，每个网络服务于对应的虚拟机实例。缺省情况下，用户通信仅限于网络内部。您的站点必须要有有效的网络才能配置云网络。

### ▼ 如何为项目配置网络

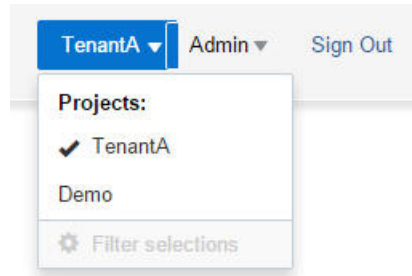
要为项目创建内部网络，应至少成为项目的成员。执行此过程不需要管理特权。

1. 在浏览器上，使用类似以下地址的 URL 登录到 Horizon 系统信息显示板：

`http://system/horizon/`

2. 通过检查缺省屏幕右上角的项目名称，确保登录到正确的项目。

以下示例显示从两个可用选项中选出了 TenantA。用户 admin 当前已登录。



3. 在左侧面板上，选择 "Project" (项目) -> "Network" (网络) -> "Networks" (网络) 选项卡，然后单击 "Create Network" (创建网络)。此时将显示 "Create Network" (创建网络) 对话框。

4. 依次单击每个选项卡以提供所需的信息。

---

注 - 单击 "Next" (下一步) 显示每个选项卡的屏幕。

---

每个选项卡会提示您提供以下信息：

- "Network" (网络) 选项卡
  - Network Name (网络名称)
  - Admin State (管理员状态) - 接受缺省值。

- "Subnet" (子网) 选项卡
  - Subnet Name (子网名称)
  - Network Address (网络地址)
  - IP Version (IP 版本)
  - Gateway IP (网关 IP) – 留空以接受缺省值。
- "Subnet Detail" (子网详情) 选项卡
  - Use DHCP (使用 DHCP) – 如果不使用 DHCP 则取消选择。
  - Allocation Pools (分配地址池)
  - DNS Server Names (DNS 服务器名)
  - Host Routes (主机路由)

在提供信息并单击 "Create" (创建) 按钮后, "Networks" (网络) 屏幕将显示网络及与其关联的子网, 如以下示例中所示:

## Networks

### Networks

<input type="checkbox"/>	Name	Subnets Associated
<input type="checkbox"/>	HR	hr_subnet 10.132.30.0/24
<input type="checkbox"/>	ENG	eng_subnet 10.132.35.0/24

Displaying 2 items

接下来的步骤 如果希望内部网络连接到公共网络, 则将该子网添加到云的外部网络。请参见[如何向内部网络提供外部连接 \[57\]](#)。

## ▼ 如何将浮动 IP 地址与项目相关联

外部网络配置中包括创建浮动 IP 地址。请参见[如何创建外部网络 \[54\]](#)。使用此过程将其中的部分 IP 地址分配给项目。

1. 在左侧面板上, 选择 "Project" (项目) -> "Compute" (计算) -> "Access & Security" (访问 & 安全) 选项卡。



2. 单击 "Floating IPs" (浮动 IP) 选项卡。
3. 单击 "Allocate IP To Project" (分配 IP 给项目) 按钮。  
此时将打开 "Allocate Floating IP" (分配浮动 IP) 对话框。
4. 从下拉式菜单中，选择从中分配浮动 IP 的池。
5. 单击对话框中的 "Allocate IP" (分配 IP) 按钮。  
一个 IP 地址已添加到 "Floating IPs" (浮动 IP) 列表中。可以分配所需任意数量的 IP 或者配额允许的 IP 数量。

## 创建和引导 VM 实例

要执行本节中的过程，必须至少成为项目的有效成员。您不需要管理特权。

### ▼ 如何创建 SSH 密钥对

1. 在浏览器上，使用类似以下地址的 URL 登录到 Horizon 系统信息显示板：  
`http://system/horizon/`
2. 通过检查缺省屏幕右上角的项目名称，确保登录到正确的项目。
3. 在左侧面板上，单击 "Project" (项目) -> "Compute" (计算) -> "Access & Security" (访问 & 安全) 选项卡。
4. 在 "Key Pairs" (密钥对) 选项卡上，确定是创建密钥对还是导入密钥对。
  - 创建密钥对。
    - a. 单击 "Create Key Pair" (创建密钥对) 按钮。
    - b. 在 "Key Pair Name" (密钥对名称) 字段上指定名称。
    - c. 单击 "Create Key Pair" (创建密钥对) 按钮。  
新密钥对会自动下载。否则，请单击提供的链接以下载密钥对。  
新密钥对会在 "Access & Security" (访问 & 安全) 面板的 "Key Pairs" (密钥对) 选项卡中列出。

- 导入密钥对。
  - a. 单击 "Import Key Pair" (导入密钥对) 按钮。
  - b. 在 "Key Pair Name" (密钥对名称) 字段上指定名称。
  - c. 从终端窗口中复制 `root` 用户的 `.ssh/id_rsa.pub` 文件的内容, 然后将其粘贴在 "Public Key" (公钥) 字段上。
  - d. 单击 "Import Key Pair" (导入密钥对) 按钮。  
新密钥对会在 "Access & Security" (访问 & 安全) 面板的 "Key Pairs" (密钥对) 选项卡中列出。

## ▼ 如何创建 VM 实例

开始之前 确保您具有 SSH 密钥对。请参见[如何创建 SSH 密钥对 \[73\]](#)。

确保已定义了内部网络。请参见[“创建项目的内部网络” \[70\]](#)。

1. 在浏览器上, 使用类似以下地址的 URL 登录到 Horizon 系统信息显示屏:  
  
`http://system/horizon/`
2. 通过检查缺省屏幕右上角的项目名称, 确保登录到正确的项目。
3. 在左侧面板上, 单击 "Project" (项目) -> "Compute" (计算) -> "Instances" (云主机) 选项卡, 然后单击 "Launch Instance" (启动云主机)。

此时将显示以下 "Launch Instance" (启动云主机) 对话框。

**Launch Instance**
✕

Details

Access & Security

Networking

**Availability Zone**

nova
▾

**Instance Name**

**Flavor**

Oracle Solaris kernel zone - tiny
▾

**Instance Count**

1
▾

**Instance Boot Source**

--- Select source ---
▾

Specify the details for launching an instance.

The chart below shows the resources used by this project

**Flavor Details**

Name	Oracle Sola...
VCPUs	1
Root Disk	10 GB
Ephemeral Disk	0 GB
Total Disk	10 GB
RAM	2,048 MB

**Project Limits**

**Number of Instances**

**Number of VCPUs**

**Total RAM**

Cancel

Launch

4. 在每个选项卡中按提示提供信息。
  - 指定以下字段的信息：
    - "Details" (详情) 选项卡
      - Instance Name (云主机名称)
      - Flavor (云主机类型) - 从下拉式菜单中选择相应的风格。如果此 OpenStack 系统是内核区域而不是裸机系统，您必须选择非全局区域风格。

- Instance Boot Source (云主机启动源) – 从下拉式菜单中选择 "Boot from image" (从镜像启动)。然后, 选择要使用的映像名称。如果此 OpenStack 系统是内核区域而不是裸机系统, 您必须选择非全局区域映像。  
风格与映像类型必须匹配。例如, 如果映像属于非全局区域, 则风格也必须是非全局区域。
  - "Access and Security" (访问 & 安全) 选项卡 – 选择要使用的密钥对。
  - "Networking" (网络) 选项卡 – 从可用网络中选择要向其附加新 VM 的网络。
5. 单击对话框底部的 "Launch" (运行) 按钮。  
将创建、安装并引导新 VM 实例。  
完成此步骤需要一段时间。
6. 将浮动 IP 地址与新 VM 实例相关联。  
您可以在安装新 VM 实例时执行这些步骤。VM 实例必须具有关联的浮动 IP 地址, 以便用户可以登录。
- a. 从 "Actions" (动作) 列的下拉式菜单中选择 "Associate Floating IP" (绑定浮动 IP)。  
将打开 "Manage Floating IP Associations" (管理浮动 IP 的关联) 对话框。
  - b. 从 "IP Address" (IP 地址) 下拉式菜单中选择地址。  
如果没有可用的 IP 地址, 请单击 + 按钮。请参见[如何将浮动 IP 地址与项目相关联 \[72\]](#)。
  - c. 选择对应于所创建的 VM 的端口。
  - d. 单击对话框底部的 "Associate" (关联) 按钮。

接下来的步骤

- 单击 "Instances" (云主机), 然后单击实例名称以查看有关该实例的详细信息, 并查看实例的控制台日志。重新装入页面以查看日志更新。
- 单击 "Volumes" (云硬盘) 以查看创建的 Cinder 卷。
- 单击 "Network Topology" (网络拓扑) 以查看云网络表示, 包括所有子网网段、虚拟路由器和活动实例。
- 单击 "Images & Snapshots" (镜像 & 快照) 以查看上载到 Glance 映像存储的统一归档文件。
- 新 VM 实例安装完成并变为 "Active" (活跃) 状态之后, 请登录到实例。以下命令使用密钥对和浮动 IP 地址以 root 用户身份登录到区域:

```
# ssh root@floating-IP-address
```

## ▼ 如何将用户添加到 VM 实例

在 Oracle Solaris 中，VM 实例使用 Oracle Solaris 区域技术以支持在云中置备虚拟机。要将用户添加到 VM 实例，需要以区域管理员身份发出命令。系统信息显示板中不支持这些步骤。因此，需要访问终端窗口。

开始之前 通过单击系统信息显示板的 "Admin" (管理员) -> "System" (系统) -> "Instances" (云主机) 选项卡，获取 VM 实例关联到的外部网络浮动 IP 地址。

1. 在终端窗口上，列出 VM 实例。

```
# zoneadm list -cv
```

云中的 VM 的名称具有前缀 instance。

2. 登录到特定的区域。

```
# zlogin zonename
```

3. 创建用户的起始目录。

```
root@zone# mkdir -p /export/home/username
```

4. 创建用户。

```
root@zone# useradd -d home-dir options
```

其中 *home-dir* 是为用户创建的目录。有关可用于 `useradd` 命令的其他选项，请参见 [useradd\(8\)](#) 手册页。

5. 要创建用户口令，请发出此命令，然后按提示操作。

```
root@zone# passwd username
```

6. (可选) 验证是否创建了口令。

```
root@zone# grep username /etc/passwd
```

7. 要创建 root 口令，请发出此命令，然后按提示操作。

```
zone# passwd root
```

8. 退出区域，然后注销。

9. 使用安全 Shell 登录到虚拟机。

```
# ssh username@floating-IP
```

其中 *floating-IP* 是与 VM 关联的浮动 IP 地址。

#### 例 4 将用户添加到 VM 实例

在此示例中，用户名 `jsmith` 被添加为 VM1 的用户。

```
# zoneadm list -cv
ID NAME                STATUS    PATH                                     BRAND  IP
0 global                running   /                                       solaris shared
6 instance-00000006    running   /system/zones/instance-00000006    solaris excl
- myzone                installed /system/zones/myzone                solaris excl

# zlogin instance-00000006
[Connected to zone 'instance-00000006' pts/3]
Last login: Wed Jan  6 14:31:18 2016 on pts/2
Oracle Corporation    SunOS 5.11    11.3    September 2015

root@VM1# mkdir -p /export/home/jsmith
root@VM1# useradd -d /export/home/jsmith -m -s /usr/bin/bash jsmith

创建用户 jsmith 时使用了 bash 作为缺省 shell。

root@VM1# passwd jsmith
New Password: password
Re-enter new Password: password
passwd: password successfully changed for jsmith

root@VM1# passwd root
New Password: password
Re-enter new Password: password
passwd: password successfully changed for root

root@VM1# exit
logout

[Connection to zone 'instance-00000006' pts/3 closed]

# ssh jsmith@10.132.10.9
```

## 管理风格

风格是 VM 实例类型，或虚拟硬件模板。风格指定一组虚拟机资源，例如虚拟 CPU 的数目、内存量以及为 VM 实例分配的磁盘空间。在 Oracle Solaris 上，风格还包括底层区域的标记：`solaris` 表示非全局区域，`solaris-kz` 表示内核区域。实例风格的一个示例为具有 16 个虚拟 CPU 和 16384 MB RAM 的内核区域。

有关风格的一般信息，请参见《*OpenStack Cloud Administrator Guide*》（《OpenStack 云管理员指南》）中的“[Flavors](#)”（风格）部分。

## 显示有关风格的信息

以云管理员身份登录到系统信息显示板时，可以通过 "Admin"（管理员）-> "System"（系统）-> "Flavors"（云主机类型）选项卡查看可用的风格。

图 5 Oracle Solaris for Oracle OpenStack 中的风格

<input type="checkbox"/>	Flavor Name	VCPU	RAM	Root Disk	Ephemeral Disk	Swap Disk	ID	Public	Metadata	Actions
<input type="checkbox"/>	Oracle Solaris kernel zone - tiny	1	2048MB	10GB	0GB	0MB	1	Yes	Yes	Edit Flavor
<input type="checkbox"/>	Oracle Solaris non-global zone - tiny	1	2048MB	10GB	0GB	0MB	6	Yes	Yes	Edit Flavor
<input type="checkbox"/>	Oracle Solaris non-global zone - small	4	3072MB	20GB	0GB	0MB	7	Yes	Yes	Edit Flavor
<input type="checkbox"/>	Oracle Solaris kernel zone - small	4	4096MB	20GB	0GB	0MB	2	Yes	Yes	Edit Flavor
<input type="checkbox"/>	Oracle Solaris non-global zone - medium	8	4096MB	40GB	0GB	0MB	8	Yes	Yes	Edit Flavor
<input type="checkbox"/>	Oracle Solaris kernel zone - medium	8	8192MB	40GB	0GB	0MB	3	Yes	Yes	Edit Flavor
<input type="checkbox"/>	Oracle Solaris non-global zone - large	16	8192MB	40GB	0GB	0MB	9	Yes	Yes	Edit Flavor
<input type="checkbox"/>	Oracle Solaris kernel zone - large	16	16384MB	40GB	0GB	0MB	4	Yes	Yes	Edit Flavor
<input type="checkbox"/>	Oracle Solaris non-global zone - xlarge	32	16384MB	80GB	0GB	0MB	10	Yes	Yes	Edit Flavor
<input type="checkbox"/>	Oracle Solaris kernel zone - xlarge	32	32768MB	80GB	0GB	0MB	5	Yes	Yes	Edit Flavor

Displaying 10 items

有关这些列的信息，请参见 [OpenStack Command-Line Interface Reference](#)（OpenStack 命令行接口参考）。

## 修改风格规范

在每个风格的 "Actions"（动作）列下，单击前三个选项之一会显示风格的属性，并可修改属性。有三个可用操作：

- Edit Flavor（编辑云主机类型）显示有关风格的信息以及可修改的属性。通过 "Flavor Access"（云主机类型访问）选项卡，可以限制哪个项目能访问风格。缺省设置为 "none"（无），即风格是公共的，所有项目都能访问。
- Modify Access（修改使用权）可直接打开风格的 "Flavor Access"（云主机类型访问）选项卡，从而可以修改访问设置。
- Update Metadata（更新元数据）允许修改风格的元数据。

修改风格的规范后，那些修改将应用于您后续创建的使用已修改风格的所有来宾。

并非所有风格修改都能在系统信息显示板上执行。例如，extra\_specs 属性的键只能通过命令行修订。属性的键是指一组区域属性，这些属性通常使用 zonecfg 命令配置，并且在 OpenStack 中受支持。

以下键在内核区域和非全局区域风格中都受支持：

- zonecfg:bootargs
- zonecfg:brand
- zonecfg:hostid
- zonecfg:cpu-arch

以下键仅在非全局区域风格中受支持：

- zonecfg:file-mac-profile
- zonecfg:fs-allowed
- zonecfg:limitpriv

有关这些区域配置属性的说明，请参见 [zonecfg\(8\)](#) 手册页。

---

注 - 并非所有区域配置属性都在 OpenStack 中受支持。

---

sc\_profile 键也只能通过命令行修改。使用此键可指定风格的系统配置文件。

要通过命令行修改风格，请使用以下语法：

```
nova flavor-key flavor action key=value [key=value . . . ]
```

*flavor*                    风格的名称或 ID。

*action*                    set 或 unset

*key=value*                *key* 是规范的名称。*value* 是该规范的新值。如果 *action* 为 unset，则仅指定 *key*。

例如，要为风格列表中的第八个风格 (Oracle Solaris kernel zone - large) 设置特定的系统配置文件，可发出以下命令：

```
$ nova flavor-key 4 set sc_profile=/system/volatile/profile/sc_profile.xml
```

有关删除和创建风格的信息，请参见《[OpenStack Admin User Guide](#)》（《OpenStack 管理员用户指南》）。

## ▼ 如何修改风格的 extra\_specs 属性

1. 为 Neutron 设置全局 shell 变量。



```

controller# export OS_USERNAME=nova
controller# export OS_PASSWORD=service-password
controller# export OS_PROJECT_NAME=service
controller# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0

```

2. 显示可用的风格。

```
controller# nova flavor-list
```

3. 记下要修改的风格的 ID。

4. 修改该风格的 extra\_specs 的键。

```
controller# nova flavor-key flavor action es-key=value
```

其中 es-key 是指 extra\_specs 属性的特定键。

5. (可选) 显示风格的属性。

```
controller# nova flavor-show flavor
```

#### 例 5 更改 zonecfg:bootargs 键

此示例说明了如何修改 ID 为 8 的 Oracle Solaris non-global zone - medium 风格的 zonecfg:bootargs 键。

为了节省空间，RTX\_Factor 和 Is\_Public 列是通过以下 nova flavor-list 样例输出编辑的。

```

controller# nova flavor-list
+-----+-----+-----+-----+-----+-----+
| ID | Name                                     | Memory_MB | Disk | Ephemeral | Swap | VCPUs |
+-----+-----+-----+-----+-----+-----+
| 1 | Oracle Solaris kernel zone - tiny      | 2048      | 10  | 0         |      | 1     |
| 10| Oracle Solaris non-global zone - xlarge| 16384     | 80  | 0         |      | 32    |
| 2 | Oracle Solaris kernel zone - small     | 4096      | 20  | 0         |      | 4     |
| 3 | Oracle Solaris kernel zone - medium    | 8192      | 40  | 0         |      | 8     |
| 4 | Oracle Solaris kernel zone - large     | 16384     | 40  | 0         |      | 16    |
| 5 | Oracle Solaris kernel zone - xlarge    | 32768     | 80  | 0         |      | 32    |
| 6 | Oracle Solaris non-global zone - tiny  | 2048      | 10  | 0         |      | 1     |
| 7 | Oracle Solaris non-global zone - small | 3072      | 20  | 0         |      | 4     |
| 8 | Oracle Solaris non-global zone - medium| 4096      | 40  | 0         |      | 8     |
| 9 | Oracle Solaris non-global zone - large | 8192      | 40  | 0         |      | 16    |
+-----+-----+-----+-----+-----+-----+

controller# nova flavor-key 8 set zonecfg:bootargs=-v
controller# nova flavor-show 8

+-----+-----+-----+-----+-----+-----+

```

Property	Value	
OS-FLV-DISABLED:disabled	False	
OS-FLV-EXT-DATA:ephemeral	0	
disk	40	
extra_specs	{"zonecfg:brand": "solaris"} u'zonecfg:bootargs': u'-v'}	<i>bootargs 已修改</i>
id	8	
name	Oracle Solaris non-global zone - medium	
os-flavor-access:is_public	True	
ram	4096	
rxtx_factor	1.0	
swap		
vcpus	8	

## 管理 VM 实例

本节介绍如何修改您在云中已创建的 VM 实例，如迁移 VM 或调整其大小以及设置 VM 引导选项。

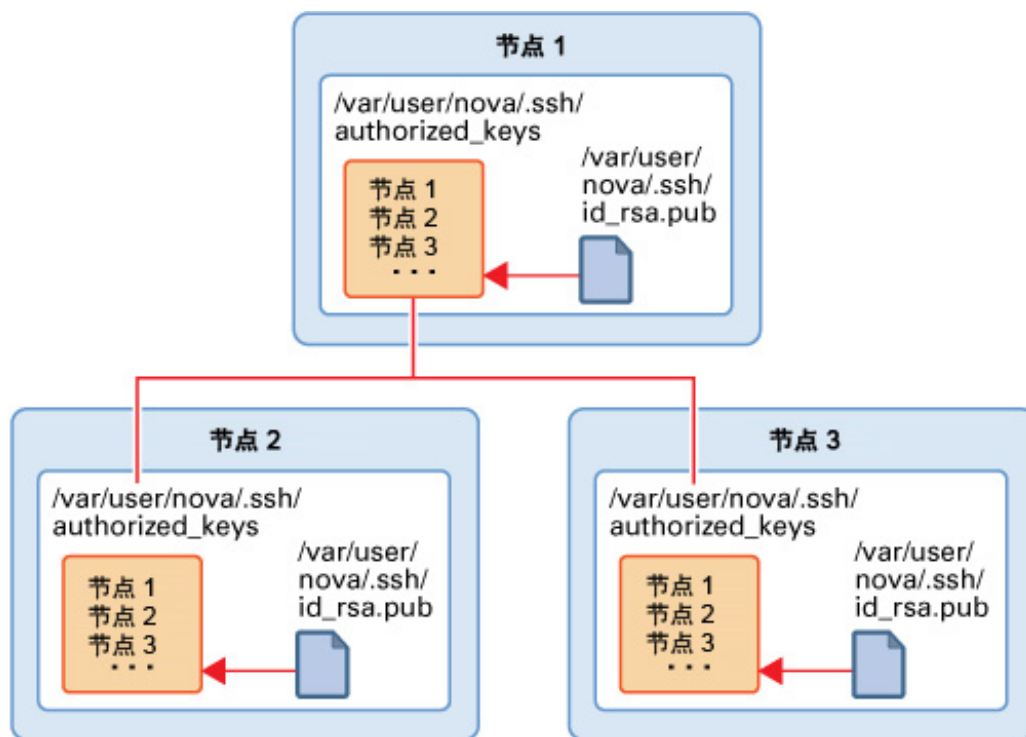
### 迁移和清除 VM 实例

在 Oracle OpenStack for Oracle Solaris 中，VM 实例是内核或非全局区域，是 Oracle Solaris 的可伸缩高密度虚拟环境。对区域实时迁移的支持已扩展到 Nova VM 实例。

可以通过使用 Horizon 服务的系统信息显示板或 nova 命令启动实时迁移。调度程序从参与节点中选择迁移的目标主机。为确保安全性，迁移机制将自动选择相应的加密算法以执行迁移。但是，通过 /etc/nova/nova.conf 文件中的参数可以选择要使用的首选加密。

有关区域实时迁移的更多信息，请参见 [Operating Systems Documentation](#)（操作系统文档）中您 Oracle Solaris 版本相应库中的关于迁移内核区域的《创建和使用 Oracle Solaris 内核区域》。另请参见 [zoneadm\(8\)](#) 和 [solaris-kz\(7\)](#) 手册页。

若要成功完成节点迁移，必须确保将每个计算节点的 SSH 密钥添加到每个节点的授权密钥文件中。因此，节点共享完全相同的授权密钥文件，如下图中所示：



通过完成以下步骤来准备 VM 实例迁移：

1. 在每个节点上，创建一个 SSH 密钥。

```
# su - nova -c "ssh-keygen -N '' -f /var/user/nova/.ssh/id_rsa -t rsa"
```

2. 将来自不同节点的所有密钥文件放到其中一个节点中的公共位置。
3. 将所有密钥组合成 `authorized_keys` 文件。

例如：

```
# cat nova(1)/id_rsa.pub nova(n)/id_rsa.pub >> /var/user/nova/.ssh/authorized_keys
```

其中 `nova(1)` 至 `nova(n)` 表示参与节点的 SSH 密钥。

4. 将 `authorized_keys` 文件分发到所有其他参与节点的 `/var/user/nova/.ssh` 目录。
5. (可选) 在每个计算节点的 `/etc/nova/nova.conf` 文件中的 `live_migration_cipher` 参数上指定要在迁移过程中使用的口令。

但是，如果希望进程自动选择合适的口令，请保持该参数的未设置状态。

要将正在运行的服务器实时迁移到其他计算机，请在设置全局 `shell` 变量后使用以下语法：

```
# nova live-migration server [host]
```

其中，*server* 可以是服务器的名称或 ID，可选 *host* 是目标服务器的名称。

如果当前实例的节点失败，或者如果 Nova 服务本身已被禁用一段时间，则可以移动或删除该实例并在其他节点上重新生成它。这样便可以恢复节点。

---

注 - 只能清除内核区域。当根设备在共享存储上时，配置中支持清除。

---

要将所有 VM 实例从一个主机实时迁移到另一个主机，请在设置全局 shell 变量后使用以下语法。

```
# nova host-evacuate-live [--target-host target] server
```

## 调整 VM 实例大小

VM 的大小通过 VM 启动时所依据的风格指示。有关创建 VM 实例的步骤，请参见[如何创建 VM 实例 \[74\]](#)。下图显示了有关样例 VM *hr\_vm1*（如 Horizon 系统信息显示板上所显示）的详细信息。

图 6 VM 实例大小

### Instances

#### Instances

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Size
<input type="checkbox"/>	hr_vm1	Solaris Non-global Zone	10.132.20.5 10.132.10.10	Oracle Solaris non-global zone - tiny
Displaying 1 item				

图中显示 *hr\_vm1* 的大小是已为 Oracle Solaris non-global zone - tiny 风格定义的大小。通过单击实例名称，可以看到有关特定风格的详细信息：

- RAM = 2 GB
- VCPU 数 = 1 个 VCPU
- 磁盘 = 10 GB

调整实例大小意味着为实例使用不同的风格。

缺省情况下，调整大小进程会在新节点上创建新调整了大小的实例。但是，如果要调整到某个非全局区域风格，则必须将调整大小进程设置为在同一个节点中完成。非全局区域需要与其原始全局区域的版本相同。调整到非全局区域风格时，会在不同的节点中创建实例，从而存在将实例放在具有不同全局区域版本的节点中的风险。要防止该风险，请在 `/etc/nova/nova.conf` 文件中编辑以下参数，从而在同一节点中创建具有新大小的实例：

```
allow_resize_to_same_host=true
```

---

注 - 该风险不涉及内核区域。因此，可以将内核区域安全地调整到其他节点。

---

## ▼ 如何调整 VM 实例大小

开始之前 了解要修改的 VM 实例的当前大小。可以从系统信息显示板获取此信息。有关示例，请参见图 6。

1. 设置全局 shell 变量。

```
# export OS_USERNAME=nova
# export OS_PASSWORD=service-password
# export OS_PROJECT_NAME=service
# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0"
```

2. 如果要调整到非全局区域风格，请按如下所示在 `/etc/nova/nova.conf` 文件中编辑以下参数：

```
allow_resize_to_same_host=true
```

3. 显示风格列表。

```
# openstack flavor list
```

4. 记下希望 VM 实例使用的新风格的 ID 号。

5. 通过引用风格的 ID 号，更改 VM 实例以使用新风格。

```
# openstack server resize --flavor flavor-ID instance-name
```

例如，假定当前实例 `hr_vm1` 的大小设置为 Oracle Solaris non-global zone - tiny 风格，该风格的 ID 号为 6。您希望将实例大小调整为 Oracle Solaris non-global zone - medium 风格，该风格的 ID 号为 8。应键入以下内容：

```
# openstack server resize --flavor 8 hr_vm1
```

在指定实例名称之前，可以插入其他选项，如 `--wait`（以等待调整大小的完成）或者 `--confirm`（以确认调整服务器大小的完成时间）。

6. （可选）在系统信息显示板上验证实例的大小是否已更改为新风格。

或者，也可以通过使用以下命令进行相同的验证：

```
# openstack server show instance-name
```

检查 flavor (风格) 字段的值是否指定了上一步中的新风格。

以下示例是一个提取的输出。实例 hr\_vm1 是在项目 TenantA 中创建的，并且大小已调整为 Oracle Solaris non-global zone - medium。

```
# export OS_USERNAME=admin
# export OS_PASSWORD=admin-password
# export OS_PROJECT_NAME=TenantA
# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0"
```

```
# openstack server show hr_vm1
```

```
+-----+-----+
| Property      | Value                                     |
+-----+-----+
| ...           |                                           |
| created       | 2016-01-26T12:38:47Z                     |
| flavor        | Oracle Solaris non-global zone - medium (8) |
| ...           |                                           |
+-----+-----+
```

## 用于 Cinder 配置和部署的选项

---

本章介绍您可以用来配置 Cinder 和 OpenStack 设置的存储组件的替代方法。

---

注 - “配置存储节点” [44] 中用来配置块存储的过程仍然是用来设置存储节点的缺省方法。本章中的选项提供了用来部署存储组件的替代方法。不过，建议不要同时使用缺省存储配置和替代配置在您的云框架上设置 Cinder。

---

本章包含以下主题：

- “部署用于存储的远程系统” [87]
- “指定计算节点的引导卷” [92]
- “使用 Cinder NFS 驱动程序” [93]
- “将 OpenStack 用于 Oracle ZFS Storage Appliance” [95]

### 部署用于存储的远程系统

在不支持 SAN 的以前的 OpenStack 发行版中，如果使用 ZFS iSCSI 驱动程序，必须将 Cinder 卷服务配置为在目标主机上运行。

借助对 Cinder 中存储区域网络 (storage area network, SAN) 的支持，您可以使用 SSH 配置多个存储主机后端以及定义其卷类型。在此类型部署中，Cinder 软件包以及所有 Cinder 服务都将仅安装在启动器主机上，该主机通常还用作计算节点。

不需要在远程目标主机上安装 OpenStack 软件包。这些主机仅向基于 COMSTAR 的启动器主机提供 LUN 磁盘。

正确部署用于存储的远程系统包含以下要求：

- 配置 `/etc/cinder/cinder.conf` 文件。
- 向指定的用户授予合适的权限配置文件。
- 手动安装其他软件包。

以下各节详细讨论了这些要求。

## 配置 cinder.conf 文件

在启动器主机上安装 Cinder 软件包后，编辑 `/etc/cinder/cinder.conf` 文件的 [DEFAULT] 部分。

请参阅以下示例，以作为理解配置文件中的定义的指南。

```
[DEFAULT]
my_ip = localhost
osapi_volume_workers = 1
auth_strategy = keystone
#num_volume_device_scan_tries = 6
os-volume-api-version = 2
scheduler_driver=cinder.scheduler.filter_scheduler.FilterScheduler

enabled_backends=zfsdriver-1, zfsdriver-2, zfsdriver-3, zfsdriver-4, zfsdriver-5

[zfsdriver-1]
volume_group=zvolumes-1
volume_driver=cinder.volume.drivers.solaris.zfs.ZFSVolumeDriver
volume_backend_name=ZFS_LOCAL
zfs_volume_base = rpool/cinder
san_is_local = True
debug=true
verbose=true
[zfsdriver-2]
volume_group=zvolumes-2
volume_driver=cinder.volume.drivers.solaris.zfs.ZFSISCSIDriver
volume_backend_name=ZFS_REMOTE
zfs_volume_base = rpool/cinder
san_ip = 10.134.13.38
san_login = user-name
san_password = password
debug=true
verbose=true
[zfsdriver-3]
volume_group=zvolumes-3
volume_driver=cinder.volume.drivers.solaris.zfs.ZFSFCDriver
volume_backend_name=ZFS_REMOTE_FC
zfs_volume_base = rpool/fc
san_ip = 10.134.13.38
san_login = user-name
san_password = password
debug=true
verbose=true
[zfsdriver-4]
volume_group=zvolumes-4
volume_driver=cinder.volume.drivers.solaris.zfs.ZFSISCSIDriver
volume_backend_name=ZFS_REMOTE
zfs_volume_base = rpool/cinder/zq
san_ip = 10.134.13.38
san_login = user-name
```



```

san_password = password
debug=true
verbose=true
[zfsdriver-5]
volume_group=zvolumes-5
volume_driver=cinder.volume.drivers.solaris.zfs.ZFSISCSIDriver
volume_backend_name=ZFS_REMOTE
zfs_volume_base = rpool/zq
san_ip = 10.134.63.182
san_login = user-name
san_password = password
debug=true
verbose=true

```

`enabled_backends` 列出启用的后端主机。在此示例中，定义了 5 台后端主机。这些后端主机在 3 台主机上使用 3 个 ZFS 驱动程序（ZFSVolumeDriver、ZFSISCSIDriver、ZFSFCDriver）。在这些主机中，一台是本地的（localhost），而其他的是远程的（10.134.13.38 和 10.134.63.182）。

`volume_backend_name` 定义具有指定名称的卷类型。此参数标识卷类型。但是，必须使用以下命令手动创建卷类型：

```

# cinder type-create vol-type
# cinder type-key vol-type set volume_backend_name=backend-name
# cinder create --display-name display --volume-type vol-type

```

这些命令分别执行以下操作：

- 创建新的卷类型。
- 为新的卷类型指定后端名称。
- 按照新的卷类型创建一个新卷。

基于以前的 Cinder 配置文件样例，键入以下命令：

```

# cinder type-create type-remote
# cinder type-key type-remote set volume_backend_name=ZFS_REMOTE
# cinder create --display-name t1 --volume-type type-remote

```

最后一个命令将基于过滤规则在名为 ZFS\_REMOTE 的一个后端上创建新卷 `t1`。

请发出一组相同的命令为 ZFS\_LOCAL 和 ZFS\_REMOTE\_FC 创建卷类型。

`zfs_volume_base` 在每个卷后端上为新的 ZFS 卷指定基础数据集。

`san_is_local`  
`san_ip` `san_login`  
`san_password` 所有 ZFS 驱动程序所基于的 SAN 驱动程序的参数。需要设置这些参数才能通过 SSH 以本地或远程方式在后端主机上发出命令。

对于每个后端，按以下两种方式之一设置 SAN 参数：

- 仅设置 `san_is_local = True`
- 一起设置 `san_ip`、`san_login` 和 `san_password`。

不要设置全部四个 SAN 参数。

```
debug=true
verbose=true
```

用于调试的可选参数。您可以不配置这些参数。

正确配置 Cinder 后，列出服务会显示每个服务状态。

```
# cinder service-list
```

Binary	Host	Zone	Status	State	Updated_at	Disabled Reason
cinder-backup	host-2	nova	enabled	up	2015-10-13 T19:22:45.000000	None
cinder-scheduler	host-2	nova	enabled	up	2015-10-13 T19:22:43.000000	None
cinder-volume	host-2	nova	enabled	down	2015-10-13 T18:31:41.000000	None
cinder-volume	@zfsdriver-1	nova	enabled	up	2015-10-13 T19:22:46.000000	None
cinder-volume	host-2 @zfsdriver-2	nova	enabled	up	2015-10-13 T19:22:47.000000	None
cinder-volume	host-2 @zfsdriver-3	nova	enabled	up	2015-10-13 T19:22:48.000000	None
cinder-volume	host-2 @zfsdriver-4	nova	enabled	up	2015-10-13 T19:22:47.000000	None
cinder-volume	host-2 @zfsdriver-5	nova	enabled	up	2015-10-13 T19:22:48.000000	None
cinder-volume	host-2 @zfsdriver-6	nova	enabled	down	2015-10-13 T18:32:55.000000	None

## 向指定的用户授予权限

要使 `san_login` 定义的用户能够使用远程主机，必须向用户授予合适的权限配置文件。以下示例展示了如何为用户创建权限配置文件。

```
# useradd -s /usr/bin/pfbash -m jdoe
```

```

# passwd jdoe password
# profiles -p "Cinder Storage management"
profiles:Cinder Storage management> set desc="Cinder Storage management on target host"
profiles:Cinder Storage management> add profiles="File System Management"
profiles:Cinder Storage management> add auths="solaris.smf.modify.stmf"
profiles:Cinder Storage management> add cmd=/usr/sbin/itadm
profiles:Cinder Storage management:itadm> set uid=0
profiles:Cinder Storage management:itadm> end
profiles:Cinder Storage management> add cmd=/usr/sbin/fcadm
profiles:Cinder Storage management:itadm> set privs=file_dac_read,sys_devices
profiles:Cinder Storage management:itadm> end
profiles:Cinder Storage management> add cmd=/usr/sbin/fcinfo
profiles:Cinder Storage management:itadm> set privs=file_dac_read,sys_devices
profiles:Cinder Storage management:itadm> end
profiles:Cinder Storage management> add cmd=/usr/sbin/stmfadm
profiles:Cinder Storage management:stmfadm> set uid=0
profiles:Cinder Storage management:stmfadm> end
profiles:Cinder Storage management> add cmd=/usr/lib/rad/module/mod_zfsmgr.so.1
profiles:Cinder Storage management:mod_zfsmgr.so.1> set privs={zone}:/system/volatile/*, \
    sys_config,sys_mount
profiles:Cinder Storage management:mod_zfsmgr.so.1> end
profiles:Cinder Storage management> add cmd=/usr/sbin/zfs
profiles:Cinder Storage management:zfs> set priv=sys_config,sys_mount
profiles:Cinder Storage management:zfs> end
profiles:Cinder Storage management> exit

# usermod -P "Cinder Storage management" jdoe

```

有关配置文件和权限的更多信息，请参阅 [Operating Systems Documentation](#)（操作系统文档）中您 Oracle Solaris 版本相应库中的《在 Oracle Solaris 中确保用户和进程的安全》。

## 将远程主机启用为目标

要将远程主机启用为目标，请执行以下步骤：

1. 在远程主机上安装 group/feature/storage-server 软件包。

```
# pkg install storage-server
```

2. 在启动器和远程主机上启用以下服务：

- svc:/system/stmf:default
- svc:/network/iscsi/target:default
- svc:/system/rad:remote

例如：

```
remote-host# svcadm enable stmf
```

```
remote-host# svcadm enable -r svc:/network/iscsi/target:default
remote-host# svcadm enable rad
```

3. 为所定义的 `zfs_volume_base` 初始化和设置访问控制列表 (Access Control List, ACL)。

例如，假设配置文件中存在 `zfs_volume_base=rpool/fc` 定义。然后，必须运行以下命令：

```
# chmod A+user:cinder:add_subdirectory:allow /rpool/fc
# zfs allow cinder clone,create,destroy,mount,snapshot rpool/fc
```

## 指定计算节点的引导卷

在使用多个存储后端的配置中或者在多机架或多 ZFS 存储 ZS3 环境中，您希望控制放置您创建的每个新实例的根卷位置。在用于 Oracle Solaris 的 OpenStack 中，通过 `/etc/nova/nova.conf` 文件的两个参数可具有此控制。

### ▼ 如何为计算实例创建根存储卷

请注意，此过程是定义多个远程后端的常规任务的一部分，如“[部署用于存储的远程系统](#)” [87] 中所述。因此，此过程使用相同的示例。

1. 在 `/etc/cinder/cinder.conf` 文件中定义 Cinder 后端。  
请参见“[配置 cinder.conf 文件](#)” [88] 中的示例，尤其是指定 `enabled_backends` 以及每个后端的卷名称。
2. 如果您的设置使用 Cinder 可用性区域，则还要在 Cinder 配置文件中定义它。  
例如：

```
[DEFAULT]
...
storage_availability_zone=cinder_az
```

3. 对于配置文件中每个已定义的后端，创建对应的卷，如以下示例所示：

```
# cinder type-create type-remote
# cinder type-key type-remote set volume_backend_name=ZFS_REMOTE
```

请发出一组相同的命令，根据需要创建其他卷类型。然后 `solariszones` 驱动程序为 Nova 实例创建实际的 Cinder 引导卷。

4. 在 Cinder 节点上，重新启动 Cinder 服务。

```
# svcadm restart cinder-volume:default
```

5. 在每个计算节点的 `/etc/nova/nova.conf` 文件中，基于 Cinder 配置文件定义以下参数中的一个或两个。

- `boot_volume_type`
- `boot_volume_az`

例如，基于以前的步骤，请编辑 Nova 配置文件，如下所示：

```
boot_volume_type=type-remote
boot_volume_az=cinder_az
```

6. 在计算节点上重新启动 Nova 服务。

```
compute-node# svcadm restart nova-compute
```

## 使用 Cinder NFS 驱动程序

Oracle Solaris 支持用于 NFS 的 Cinder 驱动程序。此驱动程序将 Cinder 基元和 API 映射到基础后端存储。具体来说，此驱动程序将 NFS 作为后端存储提供。

Cinder 驱动程序仅负责存储设备的置备以及其他管理操作。但是，驱动程序本身不在 I/O 操作的数据路径中。该驱动程序不允许实例在块级别上访问存储设备。相反，在 NFS 共享上创建文件并映射到实例。每个 NFS 文件在功能上与块设备类似。

---

注 - 当前，此驱动程序仅支持内核区域。不要在非全局区域中使用此驱动程序。

---

### ▼ 如何使用 Cinder NFS 驱动程序

**开始之前** 要使用该驱动程序，必须具有现有的 NFS 服务器以便为客户机创建 NFS 共享。配置 NFS 服务器超出了本文档的范围。请参阅其他 NFS 文档以设置服务器。

单个 NFS 服务器通常就足够了。但是，可以根据需要具有多个 NFS 服务器。

1. 将可用的 NFS 共享添加到 `/etc/cinder/nfs_shares` 文件。  
以 `host:share` 格式列出 NFS 共享。例如：

```
nfs-server-system1:/scratch/volume1
nfs-server-system2:/scratch/volume2
```

2. 编辑 `/etc/cinder/cinder.conf` 文件。

- a. 指定要使用的 NFS 驱动程序。

```
volume_driver=cinder.volume.drivers.solaris.nfs.ZfsNfsVolumeDriver
```

- b. 确保 `nfs_shares_config` 参数指定您在 [步骤 1](#) 中使用的文件。

```
nfs_shares_config=/etc/cinder/nfs_shares
```

- c. 根据需要配置与 NFS 相关的其他参数。

- `nfs_mount_attempts` – 在生成错误之前挂载 NFS 共享的最大尝试次数。
- `nfs_mount_point_base` – NFS 共享挂载点的主目录。
- `nfs_oversub_ratio` – 卷目标中已分配的卷空间和可用空间之比的最大限制。如果超过该比率，则卷目标将变为无效。
- `nfs_sparsed_volumes` – 缺省值为 True，表示将卷创建为稀疏文件。否则，将卷创建为常规文件。
- `nfs_used_ratio` – 在其上无法再将新卷分配给卷目标的底层卷的实际使用百分比。
- `nfs_round_robin` – 缺省值为 True，表示在 NFS 共享之间调度循环。如果未设置此参数，将为卷放置选择具有最多空闲空间的 NFS 共享。

3. 重新启动 Cinder 服务。

```
# svcadm cinder-volume restart
```

该服务启动后，NFS 共享的目录将添加到 NFS 共享挂载点的主目录。

4. 创建卷。

---

注 - 使用 Cinder NFS 驱动程序时，不支持创建快照。

---

- a. 定义必需的 shell 变量。

例如：

```
# export OS_USERNAME=nova
# export OS_PASSWORD=service-password
# export OS_PROJECT_NAME=service
# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0
```

- b. 创建卷。

例如：

```
# nova volume-create --display-name nfsvol 5
```

- c. (可选) 显示卷的属性。

例如：

```
# nova volume-show nfsvol
```

## 将 OpenStack 用于 Oracle ZFS Storage Appliance

本节介绍使用 Oracle ZFS Storage Appliance (ZFSSA) 作为后端存储的 OpenStack 配置。

### 关于 Oracle ZFS Storage Appliance

Oracle ZFS Storage Appliance 系列产品可为网络客户机提供高效率的文件和块数据服务，还可为系统上存储的数据提供一组数据服务，包括以下技术：

- Analytics (分析) - 允许您实时、动态监测系统行为并以图形方式查看数据。
- ZFS 混合存储池 - 允许您以一个数据分层结构来管理各种设备，例如可选闪存设备、低能耗且大容量磁盘以及 DRAM 内存。
- 支持各种硬件

Oracle OpenStack for Oracle Solaris 的 `cloud/openstack/cinder` 软件包包括 Oracle ZFSSA iSCSI Cinder 驱动程序。利用该驱动程序，可以无缝地使用设备作为 Cinder 组件的块存储资源。具体地说，利用该驱动程序，可以创建 Cinder 服务器能分配到由 Nova 服务实例化的任何虚拟机的 iSCSI 卷。要将 Oracle ZFSSA 用于存储，请先确保设备至少运行的是 ZFSSA 软件发行版的版本 2013.1.2.0。

### 使用 Oracle ZFSSA 配置 OpenStack

本节不详细介绍用于设置 Oracle ZFSSA 的过程。如需了解包括配置步骤在内的有关设备的更多信息，请参阅以下来源：

- [Oracle 帮助中心存储文档页](#)中的 Oracle ZFSSA 产品文档
- 《[Using the Oracle ZFS Storage Appliance as Storage Back End for OpenStack Cinder](http://www.oracle.com/technetwork/server-storage/sun-unified-storage/documentation/openstack-cinder-zfssa-120915-2813178.pdf)》(<http://www.oracle.com/technetwork/server-storage/sun-unified-storage/documentation/openstack-cinder-zfssa-120915-2813178.pdf>) (《使用 ZFS Storage Appliance 作为 OpenStack Cinder 的存储后端》)。此白皮书详细介绍了如何设置要在云中使用的存储设备。

在执行使用 Oracle ZFSSA 设置 OpenStack 的过程之前，请确保已经创建了将承载数据通信流量的子网。本章中的方案使用图 1 作为此起点的基础，但有一处差异。此处，存储数据通信流量承载在两个子网上，以 `net2` 和 `net3` 作为该通信流量的接口。这两个子网可确保数据存储服务的高效性能和可用性。

## 在 ZFSSA 方面

Oracle ZFSSA 使用 iSCSI 在网络上交换 SCSI 命令。要设置 iSCSI 节点之间的通信，必须提供以下信息：

- SCSI 节点 DNS 名称或 IP 地址。SCSI 节点包含启动器节点和目标节点。
- 目标节点上的 TCP iSCSI Internet 端口。缺省情况下，端口号为 3260。
- 启动器节点和目标节点的 iSCSI 限定名称 (iSCSI Qualified Name, IQN)。
- 使用质询握手身份验证协议 (Challenge Handshake Authentication Protocol, CHAP) 的可选验证信息 (CHAP 验证)

要为此预备步骤获取相关的 iSCSI 信息，请使用 Oracle Solaris `iscsiadm` 命令。

本文档不介绍设备配置步骤。有关为设备进行运营准备的过程，请参阅 Oracle 的[存储文档](#)中的相应发行版文档。

## 在 OpenStack 方面

OpenStack ZFSSA Cinder 驱动程序会创建和管理卷，其方法为：在设备上创建相关的 LUN，设置这些 LUN 的相应属性，以及管理 LUN 对要在其上使用卷的 OpenStack 计算节点和来宾实例的可见性。

在设置了 Oracle ZFSSA 后，就可以配置 ZFSSA iSCSI Cinder 驱动程序了。

## ▼ 如何配置 Oracle ZFSSA for OpenStack

此过程使用执行以下任务的工作流 `cinder.akwf`：

- 如果用户不存在，则创建用户。
- 设置执行 Cinder 驱动程序操作的角色授权。
- 如果 RESTful 服务当前处于禁用状态，则启用该服务。

开始之前 在 Oracle ZFS Storage Appliance 上配置一个池。您可以选择使用预先存在的池。

1. 通过使用以下方法之一运行工作流 `cinder.akwf`：

- 从 CLI 运行该工作流。

```
zfssa:maintenance workflows> download
zfssa:maintenance workflows download (uncommitted)> show
Properties:
    url = (unset)
    user = (unset)
    password = (unset)
```



```

zfssa:maintenance workflows download (uncommitted)> set url= "url-to-cinder.akwf-file"
                        url = "url-to-cinder.akwf-file"
zfssa:maintenance workflows download (uncommitted)> commit
Transferred 2.64K of 2.64K (100%) ... done

zfssa:maintenance workflows> ls
Properties:
                showhidden = false

Workflows:

WORKFLOW      NAME                                     OWNER SETID ORIGIN
VERSION
workflow-000 Clear locks                  root  false Oracle Corporation
1.0.0
workflow-001 Configuration for OpenStack Cinder Driver root  false Oracle Corporation
1.0.0

zfssa:maintenance workflows> select workflow-001

zfssa:maintenance workflow-001> execute
zfssa:maintenance workflow-001 execute (uncommitted)>

zfssa:maintenance workflow-001 execute (uncommitted)> set name=USER
                        name = USER
zfssa:maintenance workflow-001 execute (uncommitted)> set password=PASSWORD
                        password = PASSWORD
zfssa:maintenance workflow-001 execute (uncommitted)> commit
User openstack created.

```

请注意，对于 *user* 和 *password*，其值为在 `cinder.conf` 文件中为 `san_login` 和 `san_password` 参数定义的值。

- 从 BUI 运行该工作流。
  - a. 选择 "Maintenance" (维护) > "Workflows" (工作流)，然后使用加号图标上载新的工作流。
  - b. 单击 "Browse" (浏览) 按钮，然后选择 `cinder.akwf` 文件。
  - c. 单击 "UPLOAD" (上载) 按钮以完成工作流的上载。
  - d. 单击 "BUI Workflows" (BUI 工作流) 页面上显示的新行以运行 Cinder 驱动程序工作流。  
工作流提示输入用户名和口令。此用户名和口令还将以 `san_login` 和 `san_password` 形式用于 `cinder.conf` 文件。

2. 在 `/etc/cinder/cinder.conf` 文件中设置参数。  
在 `cinder.conf` 文件中指定以下必需属性：

注 - 以下是一个部分列表。请确保您查看并设置配置文件中使特定的设置起作用所需的所有属性。

- `volume_driver` - 确保取消对 `cinder.volume.drivers.zfssa.zfssaiscsi.ZFSSAISCSIDriver` 的注释。确保注释掉其他三个选项。
- `san_ip` - ZFSSA 管理主机的名称或 IP 地址。
- `san_login` - ZFSSA 上 Cinder 用户的用户名。
- `san_password` - ZFSSA 上 Cinder 用户的口令。
- `zfssa_pool` - 用于分配卷的池。
- `zfssa_target_portal` - ZFSSA iSCSI 目标门户 (采用 `data-IP:port` 格式)。缺省端口为 3260。
- `zfssa_project` - ZFSSA 项目的名称。如果设备上不存在该项目，则驱动程序会在启动时使用该名称创建一个项目。此项目包含该驱动程序创建的所有卷。提供的其他 ZFSSA 属性用于设置卷特征 (例如块大小) 和访问 (例如启动器、目标、安全性)。
- `zfssa_initiator_config` - 一个属性，列出多个启动器或多组驱动器。此属性替代了以前的 `zfssa_initiator_group` 参数，后者在 OpenStack Kilo 版本中已过时。

要列出多个启动器，请使用以下格式：

```
zfssa_initiator_config = {
    'init-grp1': [
        {'iqn':'iqn1', 'user':'user', 'password':'password'},
        {'iqn':'iqn2', 'user':'user', 'password':'password'}
    ],
    'init-grp2': [
        {'iqn':'iqn3', 'user':'user', 'password':'password'}
    ]
}
```

有关如何在此属性上列出启动器的特定示例，请参见例 6 “使用 `zfssa_initiator_config` 驱动程序属性”。

- `zfssa_target_interfaces` - ZFSSA iSCSI 目标网络接口。使用下面的命令查看接口：

```
zfssa:configuration net interfaces> show
Interfaces:

INTERFACE STATE CLASS LINKS   ADDR5      LABEL
e1000g0  up   ip   e1000g0  1.10.20.30/24  Untitled Interface
```

- `connection` - 按如下方式设置参数：

```
connection=mysql://cinder:service-password@controller-fqdn/cinder
```

### 3. 确保 ZFSSA iSCSI 服务处于联机状态。

如果 ZFSSA iSCSI 服务未处于联机状态，则在设备中使用 BUI 或 CLI 启用该服务。下面的示例说明如何在设备中使用 CLI：

```
zfssa:> configuration services iscsi
zfssa:configuration services iscsi> enable
zfssa:configuration services iscsi> show
Properties:
<status> = online
...
```

#### 4. 启用 Cinder 卷 SMF 服务。

```
controller# svcadm enable cinder-volume:default cinder-volume:setup
```

#### 例 6 使用 zfssa\_initiator\_config 驱动程序属性

此示例显示如何在 Cinder 配置文件中列出 zfssa\_initiator\_config 属性的多个启动器。

在示例中，在 ZFS 存储设备上创建了两组启动器（A 组和 B 组）。请在 /etc/cinder/cinder.conf 文件中列出这些启动器，如下所示：

```
zfssa_initiator_config = {
    'GroupA':[
        {'iqn':'iqn.1986-03.com.sun:01:0a43b9fdcfd5.570d7fd1', 'user':'test1',
        'password':'password1234'},
        {'iqn':'iqn.1986-03.com.sun:01:0a43b9fdcfd5.570d7fd2', 'user':'', 'password':''}
    ],
    'GroupB':[
        {'iqn':'iqn.1986-03.com.sun:01:0a43b9fdcfd5.570d7fd3', 'user':'', 'password':''}
    ]
}
```



## Neutron 部署的选项

---

在用于本文档的 3 节点 OpenStack 配置模型中，Neutron 组件与单系统图 1 中的控制器节点一起安装。本章介绍如何与系统中的其他核心组件相隔离来在内核区域内安装 Neutron 组件。本章包括以下几个主题：

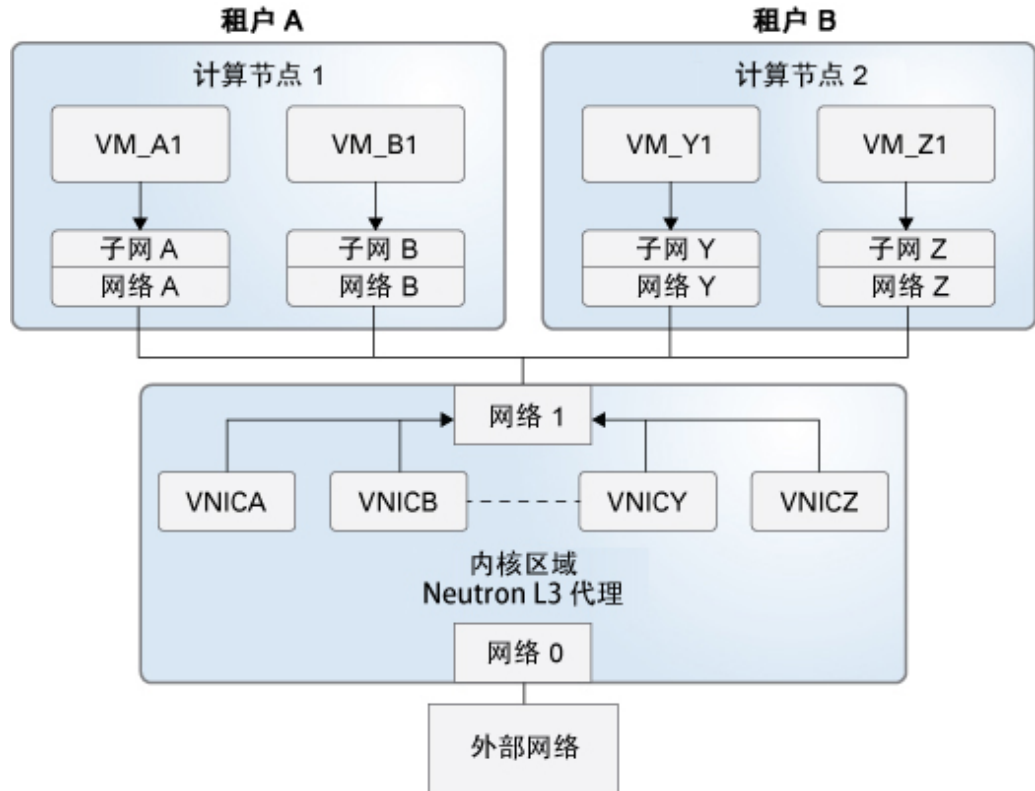
- “在内核区域中部署 Neutron” [101]
- “显示 MAC 地址和 VID 信息” [104]

### 在内核区域中部署 Neutron

在以前的 Oracle Solaris 版本中，无法动态分配地址阻止了在内核区域中安装 Neutron。新引入的区域资源属性克服了此限制。

下图显示了 Neutron 的内核区域部署。

图 7 在内核区域中部署 Neutron



在图中，VM 实例在计算节点（如 VM\_A1、VM\_B1 等）中创建，内核区域中的 L3 代理为其各自的网络配置对应的 VNIC。通过动态地址和 VID，代理可以自动管理 VM 实例的网络，因为它们由云管理员启动。

通过设置以下两个区域资源属性，启用对动态 MAC 地址和 VID 的支持：

- allowed-mac-address，已添加到 anet`mac 资源。
- allowed-vlan-ids，已添加到 anet`vlan 资源。

---

注 - 这些属性仅在 solaris-kz 标记中可用。

---

## ▼ 如何在内核区域中安装 Neutron 组件

如果希望 Neutron 组件在隔离的内核区域中而不是与其他 OpenStack 组件共享全局区域，请使用此过程。

此过程中的步骤仅侧重于与 Neutron 相关的配置。有关内核区域配置的更详细说明，请参阅相应的区域文档。

### 1. 完成创建内核区域的步骤。

有关完整说明，请参阅关于如何配置内核区域的《创建和使用 Oracle Solaris 内核区域》。此书位于 [Operating Systems Documentation](#)（操作系统文档）中您 Oracle Solaris 版本相应的库中。

### 2. 在内核区域中，分配 MAC 地址前缀的列表，L3 代理可以将这些前缀动态分配给 VNIC。

前缀长度应为 1 到 5 个八位字节，如 fa:16:3f 或 fa:80:20:21:22。

为每个前缀发出新的 add 命令。例如：

```
# zonecfg -z kernel-zone
zonecfg:kernel-zone> add anet
zonecfg:kernel-zone:anet> add mac
zonecfg:kernel-zone:anet:mac> add allowed-mac-address prefix
zonecfg:kernel-zone:anet:mac> add allowed-mac-address prefix
...
zonecfg:kernel-zone:anet:mac> end
zonecfg:kernel-zone:anet> end
zonecfg:kernel-zone>
```

### 3. 在内核区域中，定义 L3 代理可以动态分配给 VNIC 的 VLAN ID 的范围。

为每个允许的 VLAN ID 范围发出新的 add 命令。例如：

```
# zonecfg -z kernel-zone
zonecfg:kernel-zone> add anet
zonecfg:kernel-zone:anet> add vlan
zonecfg:kernel-zone:anet:vlan> add allowed-vlan-ids id-range
zonecfg:kernel-zone:anet:vlan> add allowed-vlan-ids id-range
...
zonecfg:kernel-zone:anet:vlan> end
zonecfg:kernel-zone:anet> end
zonecfg:kernel-zone>
```

也可以为 allowed-vlan-ids 属性指定关键字 any，而不是提供范围。然后 L3 代理将任何有效的 VLAN ID 分配给代理创建的 VNIC。

### 4. 在内核区域中，完成安装和配置 Neutron 的步骤。

有关说明，请参阅[如何安装并配置 Neutron \[37\]](#)。

## 显示 MAC 地址和 VID 信息

不同的命令显示不同的 MAC 地址和 VID 信息。可以使用的命令也取决于您在来宾 VM 中还是在主机上。

### 从来宾 VM 内显示

在 VM 实例内，通过使用 `dladm show-phys` 命令，可以显示可供 VM 使用的 MAC 地址和 VID 的范围。要显示这些属性，必须将 `-o` 选项与输出上所需的列一起使用。列名 `ALLOWED-ADDRESSES` 和 `ALLOWED-VIDS` 显示 MAC 地址和 VID 的范围。例如：

```
VM-instance# dladm show-phys -o link,media,device,allowed-addresses,allowed-vids
LINK  MEDIA      DEVICE  ALLOWED-ADDRESSES  ALLOWED-VIDS
net0  Ethernet    zvnet0  fa:16:3f,          100-199,
                               fa:80:20:21:22    400-498,500
```

### 从主机显示

在 VM 实例之外时，通过使用 `zonecfg info` 命令或 `zonecfg export` 命令，可以显示 MAC 地址和 VLAN ID 的范围。（可选）可以将 `-r` 选项与任一命令一起使用。

以下示例显示了从命令生成的更完整输出中提取的部分内容。

- `zonecfg info` 或 `zonecfg -r info`

```
global-zone# zonecfg -z kernel-zone -r info
anet:
...
mac:
...
allowed-mac-address: fa:16:3f
allowed-mac-address: fa:80:20:21:22
...
vlan:
...
allowed-vlan-ids: 100-199
allowed-vlan-ids: 400-498
allowed-vlan-ids: 500
...
```

- `zonecfg export` 或 `zone -r export`

```
global-zone# zonecfg -z kernel-zone -r export
add anet
```



```

...
add mac
add allowed-mac-address: fa:16:3f
add allowed-mac-address: fa:80:20:21:22
...
end
add vlan:
add allowed-vlan-ids: 100-199
add allowed-vlan-ids: 400-498
add allowed-vlan-ids: 500
end

```

区域命令显示可用 MAC 地址或 VID 的范围。

要查看实际使用的地址和 VID，请发出 `dladm show-vnic -m` 命令。在以下示例中，实际使用的地址和 ID 的信息基于 `zonecfg` 命令的上述样例输出。

```

global-zone# dladm show-vnic -m
LINK          OVER   SPEED  MACADDRESSES  MACADDRTYPES  IDS
kz1/net0      net0   1000   2:8:20:31:ab:46  random        VID:0,100-109
              2:8:20:ad:29:e8  random
              fa:80:20:21:22:00 random
              fa:80:20:21:22:ff random
              fa:16:3f:0:0:1   random
              fa:16:3f:0:0:2   random

```

输出显示代理已创建 4 个 VNIC。两个 VNIC 使用 `fa:80:20:21:22` 范围内的地址，而另两个使用 `fa:16:3f` 范围内的地址。也可以根据输出推知 4 个来宾 VM 当前存在于计算节点中。这些 VM 共使用 10 个 VID。

有关 `zonecfg` 和 `dladm` 命令的更多详细信息，请参阅 [zonecfg\(8\)](#) 和 [dladm\(8\)](#) 手册页。



## 使用 Ironic

---

本章介绍了 Oracle Solaris 中实施和支持的 Ironic。本章包括以下几个主题：

- “关于 Ironic 组件” [107]
- “安装和配置 Ironic” [108]
- “概述：包含 Ironic 的裸机部署” [112]
- “使用 Ironic 部署裸机” [114]

### 关于 Ironic 组件

前面的几章中已介绍了用于创建云的核心 OpenStack 组件。其他组件提供与云管理相关的其他服务。本章将介绍 Kilo 发行版中提供的组件 Ironic。

虽然 OpenStack 的核心组件使得可以设置虚拟机或 VM 实例，但 Ironic 提供了用于注册、置备和停用实例或服务的服务。Ironic 使用 PXE 引导和 IPMI 等常用技术来支持可以置备的广泛硬件。另外，一种可插拔的驱动程序机制使 Ironic 可以管理和支持特定于供应商的硬件。

有关 Ironic 及其所提供优势的进一步信息，请参阅 OpenStack 社区 Web 站点上适用于 Ironic 的[开发者文档](#)。

Ironic 包括三个主要组件。在 Oracle Solaris 中，这些组件是作为 SMF 服务提供的。下表列出并介绍了这些组件。

组件	说明	SMF 服务
OpenStack Ironic API 服务	此服务提供了一个 RESTful API，操作员和其他服务可以通过该 API 与受管理的裸机节点进行交互。	svc:/application/openstack/ironic/ironic-api
OpenStack Ironic 执行程序服务	通过使用参考驱动程序和特定于供应商的驱动程序执行裸机的实际置备的主控制器。执行程序服务和 API 服务使用 RPC 进行通信。	svc:/application/openstack/ironic/ironic-conductor
OpenStack Ironic 数据库服务	一种用于创建和同步 Ironic 的后端数据库的临时 SMF 服务。	svc:/application/openstack/ironic/ironic-db

## 安装和配置 Ironic

您可以将 Ironic 用作一个独立组件，而不与其他 OpenStack 服务一起使用。另外，也可以将其与其他 OpenStack 组件部署在一起，通常是在计算节点上。实际配置将取决于您如何部署 Ironic。

置备 Oracle Solaris 裸机实例需要自动化安装程序 (Automated Installer, AI)。AI 可以与 Ironic 服务位于同一节点上。另外，您还可以使用远程服务器上的 AI 执行对 Ironic 的操作。

有关配置和使用 AI 的更多信息，请参见 [Operating Systems Documentation](#)（操作系统文档）中您 Oracle Solaris 版本相应库中的相应安装书籍。

### ▼ 如何安装和配置 Ironic

**开始之前** 如果您要将 Ironic 与其余 OpenStack 组件部署在一起，请务必首先配置以下核心组件。至少，在操作 Ironic 之前必须完成 Keystone 和 Glance 配置。

同样，请确保设置 Ironic 的数据库。此过程遵循图 1 中的 3 节点参考体系结构，其中数据库安装在控制器节点上，与其余 OpenStack 数据库在一起。

#### 1. 创建 Ironic 数据库。

- a. 为 "Controller"（控制器）管理节点设置全局 shell 变量。

```
controller# export CONTROLLER_ADMIN_NODE=controller-node
```

其中 *controller-node* 可以为控制器的 IP 地址或主机名。有关设置变量的更多信息，请参见“[准备主机名、变量和口令](#)” [25]。

- b. 使用以下命令创建数据库：

```
controller# mysql -u root -p
Enter password: MySQL-root-password
mysql> create database ironic default character set utf8 default collate
utf8_general_ci;
mysql> grant all privileges on ironic.* to 'ironic'@'$CONTROLLER_ADMIN_NODE'
identified by 'service-password';
mysql> flush privileges;
mysql> quit
```

#### 2. 安装 OpenStack Ironic 软件包。

```
# pkg install ironic ironicclient rabbitmq
# pkg update stevedore stevedore-27
```

### 3. 安装 AI 服务器和管理工具。

无论您是在本地随 Ironic 安装 AI 服务器软件包还是在另一主机上远程安装，请使用相同的命令。

```
# pkg install pkg:/install/installadm
```

### 4. 如果 AI 位于远程，请在该服务器上配置 Ironic 用户。

```
remote-AI# useradd -d /var/lib/ironic -m -g 88 -u 88 \
  -P "Install Service Management" ironic
remote-AI# passwd ironic
New Password: password
Re-enter new Password: password
```

---

注 - 如果 AI 是随 Ironic 在本地安装的，则 Ironic 安装过程会自动在系统上创建 ironic 用户。

---

### 5. 为 Ironic 用户创建和管理 SSH 密钥。

- 如果 AI 随 Ironic 一起位于本地，请在该系统上发出以下命令：

```
ironic-localhost# su - ironic
ironic-localhost# mkdir /var/lib/ironic/.ssh
ironic-localhost# ssh-keygen -N '' -t rsa \
  -f /var/lib/ironic/.ssh/id_rsa
ironic-localhost# cat /var/lib/ironic/.ssh/id_rsa.pub > \
  /var/lib/ironic/.ssh/authorized_keys
```

- 如果 AI 位于远程，请执行以下步骤：

- a. 在远程 AI 服务器上，发出以下命令：

```
remote-AI# su - ironic
remote-AI# mkdir /var/lib/ironic/.ssh
remote-AI# ssh-keygen -N '' -t rsa \
  -f /var/lib/ironic/.ssh/id_rsa
remote-AI# cat /var/lib/ironic/.ssh/id_rsa.pub > \
  /var/lib/ironic/.ssh/authorized_keys
```

- b. 在 Ironic 主机上，发出以下命令：

```
ironic-localhost# mkdir /var/lib/ironic/.ssh
ironic-localhost# scp ironic@AI-server:~/.ssh/id_rsa /var/lib/ironic/.ssh
ironic-localhost# scp ironic@AI-server:~/.ssh/id_rsa.pub /var/lib/ironic/.ssh
ironic-localhost# cat /var/lib/ironic/.ssh/id_rsa.pub > \
  /var/lib/ironic/.ssh/authorized_keys
ironic-localhost# chown -R ironic:ironic /var/lib/ironic/.ssh
```

其中，AI-server 可以是 AI 服务器的 IP 地址或主机名。

6. 编辑 `/etc/ironic/ironic.conf` 文件。

注意有关在该文件中配置特定参数的以下注意事项：

<code>auth_strategy</code>	如果要将 Ironic 用作独立组件，请在 <code>DEFAULT</code> 部分下指定 <code>noauth</code> 。如果要将 Ironic 与其他 OpenStack 组件一起部署，请指定 <code>keystone</code> 。
<code>server</code>	如果自动安装程序 (Automated Installer, AI) 随 Ironic 位于本地，请在 <code>ai</code> 部分下指定 <code>localhost</code> 。如果 AI 位于远程，则指定该服务器的 IP 地址或主机名。
<code>connection</code>	在 <code>[database]</code> 部分下，指定与 MySQL 数据库的连接。
<code>glance_host</code>	如果 <code>auth_strategy</code> 参数设置为 <code>keystone</code> ，请在 <code>glance</code> 部分下指定 Glance 服务器的 IP 地址或主机名。
<code>glance_api_servers</code>	此外，还要在 <code>glance</code> 部分下指定主机名或 IP 地址及其对应的端口号，例如 <code>192.168.0.150:9292</code> 。不要指定 <code>localhost</code> 。

请使用以下样例文件指导您配置 Ironic 正常运行所需的所有参数。请注意，在该样例文件中，变量 `glance-serverIP` 表示 Glance 服务器的 IP 地址或主机名。

```
[DEFAULT]
enabled_drivers=solaris
auth_strategy= 设置取决于 Ironic 是否独立
pybasedir = /usr/lib/python2.6/vendor-packages/ironic
bindir = /usr/lib/ironic
host = ironic

[ai]
server= 设置取决于 AI 是本地的还是远程的
username=ironic
port=22
timeout=10
ssh_key_file=/var/lib/ironic/.ssh/id_rsa
deploy_interval=30

[api]
port=6385

[conductor]
api_url=http://localhost:6385/
heartbeat_timeout=60
heartbeat_interval=60
sync_power_state_interval=300
check_provision_state_interval=120

[database]
connection= mysql://ironic:service-password@$CONTROLLER_ADMIN_NODE/ironic
```

```

[solaris_ipmi]
imagecache_dirname = /var/lib/ironic/images
imagecache_lock_timeout = 60

[glance]
glance_host = glance-serverIP
glance_port = 9292
glance_protocol = http
glance_api_servers = glance-serverIP:port
auth_strategy = 对于要使用 Keystone 与 Glance 交互的 Ironic，请指定 keystone

[keystone_authtoken] 在 DEFAULT 下 auth_strategy = keystone 时配置
auth_host = localhost
auth_uri = http://$CONTROLLER_ADMIN_NODE:5000/v2.0
identity_uri = http://$CONTROLLER_ADMIN_NODE:35357
admin_user = ironic
admin_password = service-password
admin_tenant_name = tenant
signing_dir = $state_path/keystone-signing

[neutron]
auth_strategy = 对于要使用 Keystone 与 Neutron 交互的 Ironic，请指定 keystone

[oslo_messaging_rabbit]
rabbit_host=$CONTROLLER_ADMIN_NODE

```

## 7. 确保 Ironic 的 SMF 服务正在运行。

```

ironic-localhost# svcs -a | grep rabbitmq
ironic-localhost# svcs -a | grep ironic

```

如果 Ironic 处于独立模式，您可能需要手动启用这些服务。

```

ironic-localhost# svcadm enable rabbitmq
ironic-localhost# svcadm enable ironic-db
ironic-localhost# svcadm enable ironic-api ironic-conductor

```

## 8. (可选) 测试命令行实用程序。

### a. 为 Ironic 设置全局 shell 变量。

- 如果 `auth_strategy` 设置为 `noauth`，请按如下所示设置 shell 变量：

```

ironic-localhost# export OS_AUTH_TOKEN=fake-token
ironic-localhost# export IRONIC_URL=http://localhost:6385

```

- 如果 `auth_strategy` 设置为 `keystone`，请按如下所示设置 shell 变量。

如果 `Keystone` 服务位于与 `Ironic` 不同的节点中，则对于 `OS_AUTH_URL`，请指定安装了 `Keystone` 的主机名或 IP 地址。

```

ironic-localhost# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0

```

```

ironic-localhost# export OS_PROJECT_NAME=service
ironic-localhost# export OS_USERNAME=ironic
ironic-localhost# export OS_PASSWORD=service-password
ironic-localhost# export IRONIC_URL=http://localhost:6385/

```

b. 发出 Ironic 命令行。

```

# ironic driver-list
+-----+-----+
| Supported driver(s) | Active host(s) |
+-----+-----+
| solaris             | ironic         |
+-----+-----+

```

当前，solaris 是 Ironic 中唯一启用并经过测试的驱动程序。不过，您可以通过将驱动程序名称包括在 /etc/ironic/ironic.conf 文件的 [enabled\_drivers] 部分下向列表中添加驱动程序。在添加驱动程序后，需要重新启动 ironic-api 和 ironic-conductor SMF 服务。

## 概述：包含 Ironic 的裸机部署

通过 Ironic solaris 驱动程序，您可以使用 Oracle Solaris 的统一归档文件 (Unified Archive, UAR) 功能或其映像包管理系统 (Image Packaging System, IPS) 来置备裸机节点。在创建节点时，您需要通过可配置的节点元素向 solaris 驱动程序传递信息。

下表列出了用于创建节点的元素。driver\_info/archive\_uri 元素适用于通过 UAR 文件置备节点。表中的其余元素适用于使用 IPS 置备节点。

元素	说明	示例
driver_info/archive_uri	用来置备裸机的统一归档文件的 URI。	http://host.example.com/sol-11_3-x86.uar
driver_info/ai_service	要使用的 AI 服务的名称	default-x86
driver_info/publishers	IPS 发布者的列表，各个发布者以加号 (+) 分隔并使用命名格式 <i>publisher-name@origin</i> 。	solaris@http://ipkg.us.oracle.com/solarisN/dev+userland@http://my.example.repo
driver_info/fmri	要安装的 PKG FMRI 的列表，各个 FMRI 以加号 (+) 分隔。	pkg:/group/system/solaris-small-server+pkg:/cloud/openstack/nova
driver_info/install_profiles	要应用于安装环境的配置文件的 URI 的列表。各个 URI 以加号 (+) 分隔。	http://host.example.com/profile1.xml+glance://glance-image
driver_info/sc_profiles	要应用于已安装系统的系统配置文件的 URI 的列表。各个 URI 以加号 (+) 分隔。	http://host.example.com/profile1.xml+glance://glance-image
driver_info/manifest	要用来置备裸机的 AI 清单的 URI。	http://host.example.com/my-manifest.xml



元素	说明	示例
driver_info/ ipmi_address	要连接到要置备的节点的 ILOM 的串行控制台的 IP 地址或主机名。	192.168.2.200
driver_info/ ipmi_username	IPMI 连接的用户名。	root
driver_info/ ipmi_password	IPMI 连接的口令。	<i>password</i>

如果您使用 UAR 文件来置备节点，则只需要提供 driver\_info/archive\_uri 的信息。归档文件 URI 可以是以下列表中的模式之一。如果您选择使用 IPS 来置备节点，可使用相同的模式选项来指定配置文件。

- file://
- http://
- https://
- glance://

注 - 虽然 Glance 通常用于存储安装映像，但是向该存储添加配置文件是可接受的。

如果您使用 IPS 来置备节点，请首先确保至少存在一个适用于您的节点的体系结构的缺省 AI 服务，如以下示例中所示：

```
# installadm list
Service Name          Status Arch  Type Alias Aliases Clients Profiles Manifests
-----
default-i386         on   i386  pkg  yes   0     0     0     1
default-sparc        on   i386  pkg  yes   0     0     0     1
ironic-x86           on   i386  pkg  no    0     0     0     1
ironic-sparc         on   i386  pkg  no    0     0     0     1
```

```
# installadm list -vcn ironic-x86
There are no clients configured for service 'ironic-x86'.
```

```
# installadm list -vmn ironic-x86
Service Name      Manifest Name Status  Criteria
-----
ironic-x86        orig_default  default none
ironic-sparc      orig_default  default none
```

当使用 IPS 来置备节点时，指定 driver\_info/ai\_service 这一操作是可选的。如果您省略了 AI 服务的名称，则将使用适用于该节点的体系结构的缺省 AI 服务。

如果您针对 driver\_info/fmri 指定了定制软件包，则还必须针对 driver\_info/publishers 元素指定发布者。

## 使用 Ironic 部署裸机

下面的任务提供了从创建节点到实际部署的基本步骤。

有关使用 Oracle Solaris 的统一归档文件功能的更多信息，请参见 *Using Unified Archives for System Recovery and Cloning in Oracle Solaris*。此书位于您 Oracle Solaris 版本相应的库中。

### ▼ 如何通过 UAR 文件部署裸机

有关使用以下步骤的具体示例，请参见例 7 “使用 UAR 文件部署节点”。

开始之前 确保要使用的 UAR 文件已存在。

1. 创建以下变量以便于使用 Ironic 命令行实用程序。
  - IP—用来连接到节点的 ILOM 的 IP 地址。
  - USER—通常，用户为 root。
  - PASS—root 的口令。
  - HOST\_MAC—系统的 MAC 地址。

2. 创建 Ironic 节点。

```
# ironic node-create options
```

此时将显示新节点的属性，包括节点的 UUID。

3. 针对节点的 UUID 创建一个变量以便轻松引用。
4. 通过指定要安装的 UAR 位置更新节点。

```
# ironic node-update options
```

5. 为此节点创建一个关联的端口。

```
# ironic port-create options
```

6. (可选) 验证您为此节点指定的字段。

```
# ironic node-validate options
```

7. 置备节点。

```
# ironic node-set-provision-state options
```

8. (可选) 显示部署状态。

```
# ironic node-show options
```

---

注 - 如果您在置备正在进行时而不是在过程完成后运行此命令，则输出将有所不同。

---

#### 例 7 使用 UAR 文件部署节点

本示例假定：

- 用于承载节点的计算机的基本信息
  - 主机名：mynewnode.example.com
  - 体系结构：x86
  - IP 地址：1.1.1.1
  - MAC 地址：01:02:03:04:05:06
- ILOM 主机的基本信息：
  - 主机名：mynewnode-aa.example.com
  - IP 地址：2.2.2.2
  - 用户：root
  - 口令：password
- UAR 文件的名称：myuar.server/sol11-3-x86.uar

```
# export ILOM_IP=2.2.2.2
# export ILOM_USER=root
# export ILOM_PASS=password
# export HOST_MAC=01:02:03:04:05:06
```

```
# ironic node-create -d solaris -i ipmi_address=$ILOM_IP \
  -i ipmi_username=$ILOM_USER -i ipmi_password=$ILOM_PASS
```

```
+-----+
| Property      | Value                                     |
+-----+
| uuid          | 4eacbfde-4977-4d8c-8043-8cbe8f876187    |
| driver_info   | {'ipmi_address': u'2.2.2.2', 'ipmi_username': u'root', |
|               | 'ipmi_password': u'password'}          |
| extra        | {}                                       |
| driver       | solaris                                 |
| chassis_uuid | None                                    |
| properties   | {}                                       |
+-----+
```

节点已创建

```
# export NODE=4eacbfde-4977-4d8c-8043-8cbe8f876187    UUID 已保存
```

```
# ironic node-update $NODE \
  add driver_info/archive_uri=http://myuar.server/sol11-3-x86.uar
```

```
+-----+
| Property      | Value                                     |
+-----+
| instance_uuid | None                                    |
+-----+
```

```

| target_power_state | None |
| properties          | {}   |
| maintenance        | False |
| driver_info        | {'archive_uri': u'http://myuar.server/sol11-3-x86.uar', |
|                    | 'ipmi_address': u'2.2.2.2', 'ipmi_username': u'root', |
|                    | 'ipmi_password': u'password'} |
| extra              | {}   |
| last_error         | None |
| created_at         | 2014-10-03T15:38:43+00:00 |
| target_provision_state | None |
| driver             | solaris |
| updated_at        | 2014-10-03T15:53:04+00:00 |
| instance_info     |      |
| chassis_uuid      | None |
| provision_state    | None |
| reservation       | None |
| power_state       | None |
| console_enabled    | False |
| uuid              | 4eacbfde-4977-4d8c-8043-8cbe8f876187 |
+-----+-----+

```

```
# ironic port-create -n $NODE -a $HOST_MAC
```

```

+-----+-----+
| Property | Value |
+-----+-----+
| node_uuid | 4eacbfde-4977-4d8c-8043-8cbe8f876187 |
| extra     | {} |
| uuid      | 4c765ab0-2529-4463-a51b-e5544dd15a32 |
| address   | 01:02:03:04:05:06 |
+-----+-----+

```

```
# ironic node-validate $NODE
```

```

+-----+-----+
| Interface | Result | Reason |
+-----+-----+
| console   | None   | not supported |
| deploy    | True   | |
| management | True  | |
| power     | True   | |
+-----+-----+

```

```
# ironic node-set-provision-state $NODE active 节点已置备
```

```
# ironic node-show $NODE
```

```

+-----+-----+
| Property          | Value |
+-----+-----+
| instance_uuid     | None |
| target_power_state | None |
| properties        | {}   |
| maintenance       | False |
| driver_info       | {'archive_uri': u'http://myuar.server/sol11-3-x86.uar', |
|                    | 'ipmi_address': u'2.2.2.2', 'ipmi_username': u'root', |
|                    | 'ipmi_password': u'password'} |
+-----+-----+

```

```

|extra                | {}                |
|last_error           | None              |
|created_at           | 2014-10-03T15:38:43+00:00 |
|target_provision_state | deploy_complete  |
|driver               | solaris           |
|updated_at           | 2014-10-03T15:53:04+00:00 |
|instance_info        |                   |
|chassis_uuid         | None              |
|provision_state      | active           |
|reservation          | None              |
|power_state          | power on         |
|console_enabled      | False            |
|uuid                 | 4eacbfde-4977-4d8c-8043-8cbe8f876187 |
+-----+-----+

```

如果您在置备正在进行时发出了 `ironic node-show` 命令，则 `provision_state` 将指示一个不同于 `active` 的状态。

## ▼ 如何停用节点

当置备操作失败时也请使用此过程。

1. 将节点设置为已删除状态。

```
# ironic node-set-provision-state $NODE deleted
```

2. 显示节点的信息。

在下面的示例中，具有星号 (\*) 的属性指示节点的已删除状态。

```

# ironic node-show $NODE
+-----+-----+
| Property          | Value              |
+-----+-----+
|instance_uuid      | None               |
|target_power_state | None               |
|properties         | {}                 |
|maintenance        | True               |
|driver_info        | {u'archive_uri': u'http://myuar.server/sol11-3-x86.uar', |
|                   | u'ipmi_address': u'2.2.2.2', u'ipmi_username': u'root', |
|                   | u'ipmi_password': u'password'} |
|extra              | {}                 |
|last_error         | None               |
|created_at         | 2014-10-03T15:38:43+00:00 |
|target_provision_state | None               |
|driver             | solaris            |
|updated_at         | 2014-10-03T15:53:04+00:00 |
|instance_info      | {}                 |
|chassis_uuid       | None               |
|provision_state    | None               |

```

reservation	None		
power_state	power off		*
console_enabled	False		
uuid	4eacbfde-4977-4d8c-8043-8cbe8f876187		
+-----+			

3. 如果要对失败的置备过程进行故障排除，请验证节点的字段。  
为节点指定的任何 driver\_info 元素的问题都在 Reason 列下标识出来。

```
# ironic node-validate $NODE
```

Interface	Result	Reason	
console	None	not supported	
deploy	True		
management	True		
power	True		

## 使用 Heat

---

本章介绍了 Oracle Solaris 中实现和支持的 Heat。本章包括以下几个主题：

- [“关于 Heat 组件” \[119\]](#)
- [“安装 Heat” \[119\]](#)
- [“关于 HOT 模板” \[120\]](#)
- [“将 Heat 与 Cloudbase-Init 一起使用” \[121\]](#)

### 关于 Heat 组件

Heat 是 OpenStack 的编排引擎，允许根据创建的 Heat 编排模板部署云应用程序。这些模板也称为 HOT 模板。

使用 HOT 模板，可以创建不同的 OpenStack 资源类型，例如实例、浮动 IP、卷、用户等。使用模板，还可以部署高级功能，例如实例高可用性、实例自动缩放以及嵌套栈。因此，使用 Heat，所有 OpenStack 核心项目都可以扩大用户群。使用模板，Heat 为您提供一种方式，可以使用从模板传递到引擎的参数覆盖缺省资源实现。

Heat 通过 RESTful web 服务 API 提供其服务。与所有 OpenStack 应用程序一样，将使用 Python WSGI 接口并使用 Paste 一起配置应用程序。应用程序的 HTTP 端点由 Web 服务器网关接口 (Web Server Gateway Interface, WSGI) 中间件的管道组成。Heat 专门使用两个端点：端口 8004 用于 Heat API，端口 8000 用于 Heat Cloud Formation。

Heat 自己的配置由 `/etc/heat/heat.conf` 配置文件进行控制。此时，非 Solaris 特定的配置参数存在于主 Heat 配置文件中。

有关 Heat 组件的更多信息，请参阅 OpenStack 社区的 [Heat 文档](#)。

### 安装 Heat

在典型配置中，在 Keystone 所在节点上安装 Heat 服务。如果使用以下命令在节点上安装了 OpenStack，将自动包括 Heat 软件包：

```
# pkg install openstack
```

## ▼ 如何配置 Heat

开始之前 执行此任务之前，必须先按照[如何安装并配置 Keystone](#)中所述配置 Keystone。

1. 通过在 `/etc/heat/heat.conf` 中取消注释或设置参数来配置 Heat :

```
[database]
connection = mysql://heat:service-password@$CONTROLLER_ADMIN_NODE/heat

[keystone_authtoken]
auth_uri = http://$CONTROLLER_ADMIN_NODE:5000/v2.0
identity_uri = http://$CONTROLLER_ADMIN_NODE:35357
admin_user = heat
admin_password = service-password
admin_tenant_name = tenant

[oslo_messaging_rabbit]
rabbit_host=$CONTROLLER_ADMIN_NODE
```

2. 启用 heat 服务。

```
controller# svcadm enable -rs heat-api heat-db heat-engine \
heat-api-cfn heat-api-cloudwatch
```

## 关于 HOT 模板

要使用 Heat 在 OpenStack 配置中编排多个复合云应用程序，需要定义 Heat 编排模板 (Heat Orchestration Template, HOT)。HOT 模板包含您必须填写的规范。您运行进程来创建资源类型和其他高级功能时，将读取您提供的参数。

有关 HOT 模板规范及其描述，请参见 [http://docs.openstack.org/developer/heat/template\\_guide/hot\\_spec.html#hot-spec](http://docs.openstack.org/developer/heat/template_guide/hot_spec.html#hot-spec)。

有关如何编写 HOT 模板的信息，请参见 [http://docs.openstack.org/developer/heat/template\\_guide/hot\\_guide.html#hot-guide](http://docs.openstack.org/developer/heat/template_guide/hot_guide.html#hot-guide)。

要部署 HOT 模板，请使用以下命令：

```
# heat stack-create -f template stack-name
```

`template` 正在处理的模板文件的完整路径。

`stack-name` 要创建的堆栈

也可以使用格式 `-P key1=value1;key2=value2...` 在命令中插入其他参数值



以下示例显示名为 stack3 的 HOT 模板的内容，该模板创建包含子网的三个专用网络。

---

注 - 行 `heat_template_version: 2013-05-23` 必须显示在模板的顶部。

---

```
heat_template_version: 2013-05-23
description: Create a few networks.

resources:
  heat_net1:
    type:OS::Neutron::Net
    properties:
      name:heat_net1

  heat_subnet1:
    type:OS::Neutron::Subnet
    properties:
      name:heat_subnet1
      network_id: { get_resource: heat_net1 }
      cidr: 192.168.50.0/24

  heat_net2:
    type:OS::Neutron::Net
    properties:
      name: heat_net2

  heat_subnet2:
    type:OS::Neutron::Subnet
    properties:
      name:heat_subnet2
      network_id: { get_resource: heat_net2 }
      cidr: 192.168.51.0/24

  heat_net3:
    type:OS::Neutron::Net
    properties:
      name: heat_net3

  heat_subnet3:
    type:OS::Neutron::Subnet
    properties:
      name:heat_subnet3
      network_id: { get_resource: heat_net3 }
      cidr: 192.168.52.0/24
```

## 将 Heat 与 Cloudbase-Init 一起使用

Cloudbase-Init 是第三方服务，在云中自动初始化新的来宾映像，如用户创建、口令生成、脚本运行等。Oracle Solaris OpenStack 支持 Kilo 中的此服务，但是仅限于 UserData 插件。

Cloudbase-Init 软件包未包括在任何标准组软件包中。应仅在其映像明确要部署在云环境中的系统上安装此软件包。

## ▼ 如何自动初始化来宾映像

开始之前 您必须完成以下事项：

- 已创建要在云中部署其映像的系统的统一归档文件 (unified archive, UA)。确保在创建 UA 之前，在该系统中包括 cloudbase-init 软件包。  
有关创建 UA 的说明，请参阅《在 Oracle Solaris 中使用统一归档文件进行系统恢复和克隆》指南中的“使用统一归档文件”一章。此指南位于 [Operating Systems Documentation](#) (操作系统文档) 中您 Oracle Solaris 版本相应的库中。
- 在 Heat 模板的用户数据部分中，提供用户数据信息。将模板另存为 \*.yaml 文件。

1. 将包含 Cloudbase-Init 软件包的映像上传到 Glance 中。
2. 执行以下命令：

```
# heat stack-create -f yam1-template \  
-P key_name=your key name;image=image name\  
private_net=name of tenant private network stack-name
```

### 例 8 使用 Cloudbase-Init 的 Heat 模板

以下示例显示将通过 Cloudbase-Init 处理的 Heat 模板。提供的信息以粗体显示，为方便起见标记了用户数据信息。模板的名称是 test.yaml。

```
heat_template_version: 2013-05-23  
  
description: HOT template to deploy one server into an existing neutron tenant  
network and assign a floating IP address so it's routable from the public network.  
  
parameters:  
  key_name:Server1Key  
    type: string  
    description: Name of keypair to assign to server  
  image:Solaris Non-global Zone  
    type: string  
    description: Name of image to use for server  
  flavor:8  
    type: string  
    description: Flavor to use for server  
    default: 1  
  public_net:  
    type: string  
    description:  
      Name or ID of public network for which floating IP address will  
      be allocated
```

```

    default: oracle
  private_net:HR
    type: string
    description: Name or ID of private network into which server is
  deployed

resources:
  server1:
    type: OS::Nova::Server
    properties:
      name: { get_param: 'OS::stack_name' }
      image: { get_param: image }
      flavor: { get_param: flavor }
      key_name: { get_param: key_name }
      networks:
        - port: { get_resource: server1_port }
      user_data_format: RAW
----- 用户数据部分开始 -----
      user_data:
        str_replace:
          template: |
            #!/bin/ksh
            print "This is a test."

  server1_port:
    type: OS::Neutron::Port
    properties:
      network: { get_param: private_net }

  server1_floating_ip:
    type: OS::Neutron::FloatingIP
    properties:
      floating_network: { get_param: public_net }
      port_id: { get_resource: server1_port }

outputs:
  server1_private_ip:
    description: IP address of server1 in private network
    value: { get_attr: [ server1, first_address ] }
  server1_public_ip:
    description: Floating IP address of server1 in public network
    value: { get_attr: [ server1_floating_ip, floating_ip_address ] }

```

要部署模板，请键入以下内容：

```

# heat stack-create -f test.yaml -P key_name=Server1Key \
  -P image=Solaris Non-global Zone\
...-P flavor=8
  -P private_net=HR teststack

```

请注意，命令中每个选项的特定值来自 test.yaml 模板文件中的信息。公共网络是 oracle，这是缺省值。



## OpenStack 故障排除

---

本章介绍与当前发行版相关的 OpenStack 问题。它还提供了故障排除技巧和解决方法来解决您在配置期间可能会遇到的基本问题。本章包含以下主题：

- “获取命令行帮助” [125]
- “已知限制” [126]
- “检查日志文件” [127]
- “调查和解决问题” [129]
- “有关调试的一般提示和技巧” [135]
- “有用的站点” [135]

### 获取命令行帮助

在 OpenStack 中，命令对应于 OpenStack 组件。例如，nova 命令适用于计算操作，cinder 适用于存储，neutron 适用于网络。

要获取有关使用这些命令的帮助，如正确的语法、支持的子命令、可能的选项等，请使用 `command-component help` 命令，例如 `nova help` 或 `neutron help`。可以使用 `grep` 命令过滤可与 `command-component` 命令一起使用的子命令列表。例如，要列出与路由器相关的 neutron 子命令，可键入以下命令：

```
# neutron help | grep router
l3-agent-list-hosting-router  List L3 agents hosting a router.
l3-agent-router-add          Add a router to a L3 agent.
l3-agent-router-remove       Remove a router from a L3 agent.
net-gateway-connect         Add an internal network interface to a router.
router-create                Create a router for a given tenant.
router-delete                Delete a given router.
router-gateway-clear         Remove an external network gateway from a router.
router-gateway-set           Set the external network gateway for a router.
router-interface-add         Add an internal network interface to a router.
router-interface-delete      Remove an internal network interface from a router.
router-list                  List routers that belong to a given tenant.
router-list-on-l3-agent      List the routers on a L3 agent.
router-port-list             List ports that belong to a given tenant, with specified
router.
router-show                  Show information of a given router.
```

```
router-update          Update router's information.
```

然后，要获取有关子命令（例如，用于标识云中路由器的 `router-list`）的特定详细信息，可键入以下命令：

```
# neutron help router-list
```

由于 OpenStackClient (OSC) 的实现所致，基于组件的某些命令已过时。相反，将 `openstack` 作为主命令与相应的子命令一起使用。有关 OSC 的简要说明，请参见“[OpenStackClient 的实现](#)” [15]。

要获取有关 `openstack` 命令及其子命令的信息，请使用以下命令之一：

- `openstack help subcommand`
- `openstack --help`

键入不带任何子命令的 `openstack` 将切换到交互模式，可以在其中键入 `help [subcommand]` 以了解信息。要退出交互模式，请键入 `quit`。

有关 OSC 的更多信息，请参见 <http://docs.openstack.org/developer/python-openstackclient/index.html>。

有关以前的命令及其在 OSC 中等效项的列表，请参见[附录 B, OpenStackClient 命令](#)。

## 已知限制

以下是 Oracle Solaris 中的 OpenStack (Kilo) 的已知问题：

- 仅完全支持运行 Oracle Solaris 的 Nova 节点，因为 Neutron 仅针对网络虚拟化支持单个插件。
- 当前，非全局区域中不支持 Cinder 卷附件。
- VM 实例必须正在运行当前 Oracle Solaris 版本。
- 不支持 Cinder 备份。

安装 `cinder` 软件包时会安装 `cinder-backup` 服务。不过，在如“[配置存储节点](#)” [44] 所述的缺省 Cinder 部署中，该服务当前对 `backup_volume` 不起作用。

- 在系统信息显示板的 "Launch Instance"（启动云主机）对话框中，"Instance Boot Source"（云主机启动源）仅支持 "Boot from image"（从镜像启动）。在 "Project"（项目）> "Images & Snapshots"（镜像 & 快照）> "Actions"（动作）菜单中，不支持 `UploadToImage`。
- 不支持将 VXLAN 数据链路作为 `/etc/neutron/l3_agent.ini` 文件中 `external_network_dataink` 选项的值。如果将 VXLAN 数据链路设置为 `external_network_dataink` 选项的值，Neutron L3 代理将无法在外部网络上创建和激活 (plumb) VNIC。
- 必须使用命令行修改项目的网络资源的配额。

无法从 Horizon 修改网络资源的配额。可以使用 Horizon 系统信息显示板创建项目或修改现有项目的非网络资源。要修改项目的网络、子网、端口、路由器或浮动 IP 地址的配额，必须使用 `neutron quota-update` 命令。

即使在修改非网络资源时，也会看到以下错误消息。您可以忽略此消息。与此消息相反，已应用非网络资源的配额。

```
Error: Modified project information and members, but unable to modify project quotas.
```

- SMF 和 OpenStack 报告的服务状态可能不同。

以下示例显示尽管 SMF 将 nova-cert 服务显示为 online，但在 OpenStack 中已禁用该服务：

```
root@c190-133:~# nova service-disable c190-133 nova-cert
+-----+-----+-----+
| Host    | Binary  | Status  |
+-----+-----+-----+
| c190-133 | nova-cert | disabled |
+-----+-----+-----+
root@c190-133:~# svcs nova-cert
STATE          STIME      FMRI
online         21:14:11  svc:/application/openstack/nova/nova-cert:default
```

## 检查日志文件

SMF 服务和各个 Oracle Solaris 进程会生成日志文件，您可以在其中查找错误消息或收集有关在屏幕上显示的消息的更多信息。SMF 服务日志文件包含有价值的调试信息。

因为 OpenStack 通常是跨多个系统安装的，因此需要检查的日志文件也在不同的位置。有关更系统化的故障排除，请查阅每个节点的日志。

有关修复 SMF 服务问题的一般帮助，请参见 [Operating Systems Documentation](#)（操作系统文档）中您 Oracle Solaris 版本相应的库中关于解决服务问题的《在 Oracle Solaris 中管理系统服务》。

要查看服务日志，请确保拥有恰当的授权。采用相应的 RBAC 配置文件查看 OpenStack 服务日志文件或使用 `pfedit` 命令修改 OpenStack 服务配置文件。可以指定以下配置文件：

- OpenStack Block Storage Management (OpenStack 块存储管理)
- OpenStack Compute Management (OpenStack 计算管理)
- OpenStack Identity Management (OpenStack 身份管理)
- OpenStack Image Management (OpenStack 映像管理)
- OpenStack Network Management (OpenStack 网络管理)
- OpenStack Object Storage Management (OpenStack 对象存储管理)
- OpenStack Management (OpenStack 管理)

要进行故障排除，请使用以下常规命令：

- 找出特定节点上正在运行哪些 OpenStack 服务：

```
# svcs -a | grep openstack
```

- 列出哪些服务可能处于维护模式：

```
# svcs -x
```

```
svc:/application/openstack/swift/swift-replicator-rsync:
    default (OpenStack Swift Replication Service)
State: maintenance since Fri May 22 04:06:11 2015
Reason: Start method exited with $SMF_EXIT_ERR_FATAL.
    See: http://support.oracle.com/msg/SMF-8000-KS
    See: rsync(1)
    See: rsyncd.conf(5)
    See: /var/svc/log/application-openstack-swift-swift-replicator-rsync:default.log
Impact: This service is not running.
```

如果某个服务处于维护模式，请检查服务日志文件。

- 确定特定 OpenStack 服务的日志：

```
# svcs -L openstack-service
```

例如：

```
# svcs -L neutron-server
```

```
/var/svc/log/application-openstack-neutron-neutron-server:default.log
```

在拥有恰当授权的情况下，您可以组合选项（如 -Lv）来列出和查看服务的日志。

- 要立即确定在特定日志中记录的错误实例，可使用通用的 UNIX 命令，如 `grep`。

```
# grep keyword `svcs -L openstack-service`
```

可以搜索 `error`、`warning` 及其他重要关键字实例来直接阅读错误消息。

- 要在解决网络问题时检查 EVS 属性，可使用各种 `evsadm` 子命令，如 `evsadm show-prop`。

以下日志通常包含对故障排除有用的信息：

- `nova-compute`
- `nova-scheduler`
- `cinder-scheduler`
- `neutron-server`

除了 SMF 服务日志文件，还可以检查 `/var/log` 目录中的日志。就像其他 Oracle Solaris 进程一样，OpenStack 服务还会在 `/var/log/openstack-service` 目录中生成自己的日志文件。



例如，OpenStack 映像存储日志文件位于 `/var/log/glance`。创建和引导 VM 实例的问题可能会记录在 `/var/log/zones` 目录中。消息日志存储为 `/var/log/rabbitmq/rabbit@hostname.log`。

大多数 OpenStack 配置文件位于 `/etc` 目录中的 OpenStack 服务名称下。例如，OpenStack 网络配置文件位于 `/etc/neutron`。Horizon 的配置文件位于 `/etc/openstack_dashboard`。Nova 的这些配置文件位于 `/etc/nova` 中，依此类推。可以使用这些文件排除特定服务的故障：在服务的配置文件中设置或取消注释以下参数：

- `debug=true`
- `verbose=true`

通过这些参数，可以看到更多受该配置文件影响的操作的输出。请参见 [OpenStack 文档站点上 http://www.oracle.com/technetwork/articles/servers-storage-admin/getting-started-openstack-os11-2-2195380.html](http://www.oracle.com/technetwork/articles/servers-storage-admin/getting-started-openstack-os11-2-2195380.html) 和《*OpenStack Configuration Reference*》（《OpenStack 配置参考》）中“Common Configuration Parameters for OpenStack”（“OpenStack 的常见配置参数”）中的配置选项表。

---

注 - 单个 OpenStack 服务命令也可使用 `--debug` 选项。此选项与在配置文件中设置 `debug=true` 等效。

---

## 调查和解决问题

本节介绍了一些在安装和配置 OpenStack 时可能会遇到的问题。

以下示例显示了与系统信息显示板相关的错误。

```
Error: Unauthorized: Unable to retrieve usage information.
Error: Unauthorized: Unable to retrieve quota information.
Error: Unauthorized: Unable to retrieve project list information.
Error: Unauthorized: Unable to retrieve instance list information.
```

这些消息可能指示 RSA 主机密钥已发生更改，但尚未传播到所有组件。有关配置 RSA 密钥的更多信息，请参阅[如何安装并配置 Neutron](#)和[如何配置计算节点](#)。

以下错误报告可能包括在 `nova-scheduler` 日志中。

```
controller# grep error `svcs -L nova-scheduler`
2014-12-03 12:49:19.271 3475 TRACE
nova.openstack.common.rpc.common error: [Errno 32] Broken pipe
```

在刷新某个 OpenStack 服务但不刷新其他服务时，通常会报告中断的管道错误。如果在某个节点上的配置文件中做了任何更改，请刷新该节点上的所有服务。以下命令可重新启动需要刷新的联机服务。

```
controller# svcs `*openstack*` | grep online \
```

```
| awk -e '{print $3}' | xargs svcadm restart
```

缺乏资源也可能导致错误。在创建 VM 实例的过程中，nova-compute 日志可能会显示如下所示的消息：

```
[abc-123-def-456] Build of instance
abc-123-def-456 aborted: Image
xyz-987-uvw-654 is unacceptable: Image query failed.
Possibly invalid or corrupt. Log file location: controller:/tmp/archive_log.4249
```

而且，该日志还会指示 out of space/storage（空间/存储不足）。使用 top 命令可查看系统上的资源。如果系统内存小于 1 GB，则可能需要增加内存。

## 网络创建

在配置内部网络及其组件时，控制台屏幕上可能会不断显示以下消息：

```
To: root@controller.mydomain.com
From: neutron@controller.mydomain.com
Subject: *** SECURITY information for controller ***
Content-Length: 143

controller: datetime : neutron user NOT in sudoers; TTY = unknown ; PWD=var/lib/neutron;
user=ROOT ; COMMAND=command:
```

为防止生成此消息，请编辑 /etc/neutron/neutron.conf 文件，如下所示：

1. 对用于 root\_helper 选项的行取消注释。
2. 确保该参数设置为 none。

```
root_helper =
```

## VM 实例安装和配置

本节中讨论的问题专门与 VM 实例相关。

### VM 实例处于错误状态

VM 实例可能处于错误状态的一个原因是您尝试安装其体系结构与主机系统不同的 VM 实例。在这种情况下，您可能不会收到专门指明体系结构不匹配的错误消息。要避免该问题，请确保在将映像上载到 glance 映像存储时正确设置映像的 architecture 属性。使用 Horizon 上载映像时，必须在上载后设置映像的属性。也可以使用命令行在一个 glance image-create 命令中上载映像并设置属性值。有关示例，请参见[“为 Glance 系统信息库准备映像” \[59\]](#)。

## VM 实例属性值与区域属性值不匹配

OpenStack 报告的有关 VM 实例的部分信息与 Oracle Solaris 报告的有关相应区域的信息不匹配。Horizon 中显示的或 nova 命令显示的信息可能与 zoneadm 命令或其他 Oracle Solaris 命令显示的信息不匹配。

Name (名称)	Horizon 中显示的或 nova list 命令显示的 VM 实例的名称是您在创建实例时指定的名称，如 example-instance。zoneadm list 命令显示的区域的名称与 instance-00000001 类似。使用 nova show 命令确定哪个区域与哪个 VM 实例相关联。在 nova show 输出中，OS-EXT-SRV-ATTR:instance_name 属性的值是区域的名称，name 属性的值是 VM 实例的名称。
UUID	Horizon 中显示的或 nova show 命令显示的 VM 实例的 UUID 与 zoneadm list -p 命令显示的同一区域的 UUID 不匹配。zoneadm 命令显示的 UUID 是与用于 Nova 的标识符不同的标识符。
CPU	Horizon 中显示的 VM 实例的 VCPU 数量是仅虚拟化至该实例可以使用的部分 CPU 数量的上限。该数量不提供上限实例内的可观察性。psrinfo 命令报告分配给区域的专用 CPU。
Memory (内存)	当您登录到 VM 实例时，Horizon 中显示的 VM 实例的内存量可能与 prtconf 命令显示的内存量不同。Horizon 显示用于创建 VM 实例的风格所指定的内存量。prtconf 命令报告所有系统内存。
Storage (存储)	Horizon 中显示的 VM 实例的存储量可能与您登录到该 VM 实例时显示的存储量不同，除非该 VM 实例是使用共享存储上的区域 (Zones on Shared Storage, ZOSS) 的非全局区域。

## 凭证问题

在某些情况下，可能会遇到与导致无法发出服务命令的不正确凭证相关的错误消息。例如，在发出 glance 命令时，可能会生成以下错误消息：

```
Invalid OpenStack Identity credentials.
```

此消息的根本原因可能每次都不一样。因此，必须检查日志来限定可能的原因。以 glance 服务为例，请看 Glance SMF 服务日志的内容。/var/log/glance/api.log 可能会报告如下内容：

```
WARNING keystonemiddleware.auth_token [-] Authorization failed for token
```

如果在 Glance 配置文件中设置了 Debug = True 和 Verbose = True，则在 /var/svc/log/application-openstack-glance-glance-api:default.log 中会提供更多详细信息，如下所示：

```
DEBUG keystonemiddleware.auth_token [-] Received request from user:
  user_id None, project_id None, roles None service: user_id None,
  project_id None, roles None
__call__ /usr/lib/python2.7/vendor-packages/keystonemiddleware/auth_token.py:821
```

您可以浏览以下区域以解决问题。

- 检查服务配置文件以确保正确地定义了相关参数。
- 检查服务的全局 shell 变量是否正确。例如，对于 Glance 服务，应设置以下变量：
  - OS\_USERNAME=glance
  - OS\_PASSWORD=service-password
  - OS\_PROJECT\_NAME=service
  - OS\_AUTH\_URL=http://\$CONTROLLER\_ADMIN\_NODE:5000/v2.0

如果命令仍然失败并报告相同的错误消息，则可能需要重新创建服务用户以生成新的凭证。请研究以下示例：

```
# export OS_USERNAME=admin
# export OS_PASSWORD=service-password
# export OS_PROJECT_NAME=project
# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0
```

```
# openstack user list
```

从此命令的输出中，记录损坏的服务用户的 ID 号。您将删除该用户，然后使用正确的凭证重新创建它：

```
# openstack user delete user-ID

# openstack user create --name glance --password service-password
# openstack user role add --user=glance --project=service --role=admin
```

## 与 Horizon 相关的问题

在启动 VM 实例后，Horizon 系统信息显示板可能无法访问并且会显示 404 Not Found 错误消息。Apache 服务的日志将包含以下条目：

```
Oct 13 16:13:53 Executing start method ("/lib/svc/method/http-apache24 start"). ]
Apache version is 2.4
(125) Address already in use: AH000/2: make_sock: could not bind to address [::]:80
Oct 13 16:13:55 Method "start" exited with status 0. ]
```

该日志指示无法将地址绑定到端口 80，因为该端口正忙。

从 Kilo 版本开始，OpenStack 使用 Apache 2.4，而非以前的 OpenStack 版本的 Apache 2.2。确保正确的 Apache 版本已启用且在该端口上侦听。

要释放该端口，请执行以下步骤：

1. 获取当前在该端口上侦听的进程 ID。

```
# netstat -uan -f inet | grep "*.80"
```

此过程假定设置使用 IPv4 地址并且占有端口 80 的进程正在所有这些地址上侦听。如果进程在侦听 IPv6 通信，则此命令可能不会提供任何结果。

2. 根据进程 ID，识别实际进程或服务。您可以使用以下命令之一：

```
# svcs -p | egrep "online|pid http" | ggrep -B1 pid
```

或者

```
# ps -lf -p pid
```

3. 如果错误的 Apache 版本正在使用该端口，请禁用该服务。
4. 为 Kilo 版本启用正确的 Apache 版本。

如果正确的 Apache 版本处于维护模式，则在启用服务前请先清除服务。

以下示例展示了如何释放端口 80 并切换到正确的 Apache 版本。

```
# netstat -uan -f inet | grep "*.80"
*.80      *.*      root      5560 httpd    0      0 128000    0 LISTEN
*.8080    *.*      webservd  1124 java     0      0 128000    0 LISTEN
*.8009    *.*      webservd  1124 java     0      0 128000    0 LISTEN

# svcs -p | egrep "online|5560 http" | ggrep -B1 5560
online    Aug_31   svc:/network/http:apache22
          Sep_09   5560 httpd

# svcadm disable apache22
# svcadm clear apache24
# svcadm enable apache24
```

## 可伸缩性问题

缺省情况下，RabbitMQ 对文件描述符有 255 的数量限制。此限制很容易阻止您在创建几个计算节点后进一步对云进行扩展。要避免此阻止，请在 `/etc/rabbitmq/rabbitmq-env.conf` 文件中增大该限制值：

```
# Increase soft limit on file descriptors for RabbitMQ
ulimit -n 8192
```

## 中断网络

如果在网络节点上配置 Neutron 时遇到问题并可能需要中断配置来重新开始，请遵循此过程。根据您需要何时开始撤消配置，遵循该过程中所提供的顺序。

## ▼ 如何删除 Neutron 上的网络配置

1. 在 Horizon 系统信息显示板上执行此步骤。

- a. 取消关联所有浮动 IP 地址。
- b. 删除所有浮动 IP 地址。

2. 在终端窗口上，键入以下命令：

```
# neutron router-gateway-clear router-ID external-network-ID
```

```
# neutron router-interface-delete router-ID subnet-ID
```

- a. 要删除路由器网关接口，请键入以下命令：

```
# neutron router-gateway-interface-delete router-ID external-network-ID
```

- b. 要删除剩余的路由器接口，请键入以下命令：

```
# neutron router-interface-delete router-ID subnet-ID
```

3. 在 Horizon 系统信息显示板上执行以下操作。

- a. 终止所有 VM 实例。
- b. 删除子网。  
如果删除子网时出现问题，请参阅[如何删除 Vport \[134\]](#)。
- c. 删除网络。

## ▼ 如何删除 Vport

如果出现阻止您删除子网的问题，请使用此过程。

1. 确定当前使用的 vport。

```
# evsadm
```

2. 重置正在使用的 vport。

```
# evsadm reset-vport vport
```

3. 删除 vport。

```
# evsadm remove-vport vport
```

## 有关调试的一般提示和技巧

以下一般提示可帮助您在解决 OpenStack 中的问题时入门：

- 在各个配置文件中设置 `debug = true` 和 `verbose = true` 可帮助诊断问题。在有些配置文件上，可以查找某个被注释掉的占位符值，您可以将其切换为启用详细日志记录。
- 在更改组件的配置文件时，请重新启动组件的服务以使更改生效。
- 使用 `tail -30 `svcs -L service-name`` 命令可获得 SMF 日志中的信息。如果启用了调试和详细日志记录，则可能需要增加为 `tail` 命令指定的行数。
- Horizon 进程经过 Apache。因此，要诊断 Horizon，请在 `/etc/openstack_dashboard/local_settings.py` 文件中启用 `debug = True`。然后，会在 Web 页上生成 Django 错误。
- Nova 组件是基于 Oracle Solaris 区域构建的。因此，还可以参阅 `/var/log/zones` 中的日志来解决 Nova 问题。

## 有用的站点

请参阅以下站点以获取解决不同 OpenStack 问题的故障排除提示。

- <https://blogs.oracle.com/openstack/>
- <https://ask.openstack.org/en/questions/>
- <https://raymii.org/s/tags/openstack.html>







## 常用 OpenStack 配置文件和服务

---

本附录列出核心 OpenStack 组件以及 OpenStack SMF 服务的典型配置文件。

### 配置文件

#### Cinder 文件

- `/etc/cinder/api_paste.ini`
- `/etc/cinder/cinder.conf`

#### Glance 文件

- `/etc/glance/glance-api.conf`
- `/etc/glance/glance-cache.conf`
- `/etc/glance/glance-registry.conf`
- `/etc/glance/glance-scrubber.conf`
- `/etc/glance/glance-registry-paste.ini`
- `/etc/glance/glance-api-paste.ini`

#### Keystone 文件

- `/etc/keystone/keystone.conf`
- `/etc/keystone/keystone-paste.ini`

#### Neutron 文件

- `/etc/neutron/neutron.conf`

- /etc/neutron/dhcp\_agent.ini
- /etc/neutron/l3\_agent.ini
- /etc/neutron/api-paste.ini
- /etc/neutron/metadata\_agent.ini
- /etc/neutron/plugins/evs/evs\_plugin.ini

## Nova 文件

- /etc/nova/nova.conf
- /etc/nova/api\_paste.ini

## Horizon 文件

- /etc/openstack\_dashboard/local\_settings.py
- /etc/apache2/2.4/httpd.conf
- /etc/apache2/2.4/conf.d/openstack-dashboard-http.conf  
或者  
/etc/apache2/2.4/conf.d/openstack-dashboard-tls.conf

## Swift 文件

- /etc/swift/swift.conf
- /etc/swift/account-server.conf
- /etc/swift/container-server.conf
- /etc/swift/object-server.conf
- /etc/swift/proxy-server.conf
- /etc/swift/rsyncd.conf

## OpenStack SMF 服务

### Cinder

svc:/application/openstack/cinder/cinder-db:default

```
svc:/application/openstack/cinder/cinder-backup:default
svc:/application/openstack/cinder/cinder-scheduler:default
svc:/application/openstack/cinder/cinder-api:default
svc:/application/openstack/cinder/cinder-volume:setup
svc:/application/openstack/cinder/cinder-volume:default
```

## Glance

```
svc:/application/openstack/glance/glance-db:default
svc:/application/openstack/glance/glance-registry:default
svc:/application/openstack/glance/glance-scrubber:default
svc:/application/openstack/glance/glance-api:default
```

## Keystone

```
svc:/application/openstack/keystone:default
```

## Neutron

```
svc:/application/openstack/neutron/neutron-server:default
svc:/application/openstack/neutron/neutron-dhcp-agent:default
svc:/application/openstack/neutron/neutron-metadata-agent:default
svc:/application/openstack/neutron/neutron-l3-agent:default
```

## Nova \*

```
svc:/application/openstack/nova/nova-objectstore:default
svc:/application/openstack/nova/nova-consoleauth:default
svc:/application/openstack/nova/nova-novncproxy:default
svc:/application/openstack/nova/nova-api-metadata:default
svc:/application/openstack/nova/nova-api-ec2:default
svc:/application/openstack/nova/nova-api-osapi-compute:default
svc:/application/openstack/nova/nova-conductor:default
svc:/application/openstack/nova/nova-cert:default
svc:/application/openstack/nova/nova-compute:default
svc:/application/openstack/nova/nova-scheduler:default
```

\* 也可能会列出其他 Nova 服务，具体取决于计算节点设置。

## Swift

```
svc:/application/openstack/swift/swift-object-expirer:default
```

```
svc:/pplication/openstack/swift/swift-account-reaper:default
svc:/application/openstack/swift/swift-container-replicator:default
svc:/application/openstack/swift/swift-account-replicator:default
svc:/application/openstack/swift/swift-object-auditor:default
svc:/application/openstack/swift/swift-container-updater:default
svc:/application/openstack/swift/swift-container-sync:default
svc:/application/openstack/swift/swift-object-updater:default
svc:/application/openstack/swift/swift-account-auditor:default
svc:/application/openstack/swift/swift-replicator-rsync:default
svc:/application/openstack/swift/swift-container-auditor:default
svc:/application/openstack/swift/swift-object-replicator:default
svc:/application/openstack/swift/swift-container-reconciler:default
svc:/application/openstack/swift/swift-container-server:default
svc:/application/openstack/swift/swift-object-server:default
svc:/application/openstack/swift/swift-proxy-server:default
svc:/application/openstack/swift/swift-account-server:default
```

## OpenStackClient 命令

---

### OpenStackClient (OSC)

OpenStackClient 将用于计算、身份、映像、对象存储和块存储 API 的完整命令集一起放在具有统一命令结构的单个 shell 中。新命令集将 `openstack` 用作主命令，而不是在以前 OpenStack 版本中使用的组件名称。客户机的实现从 Kilo 版本开始。本附录将以前的 OpenStack 子命令映射到新客户机。

---

注 - 等效项的列表是一个部分列表。

---

要获取有关 `openstack` 命令及其子命令的信息，请使用以下命令之一：

- `openstack help subcommand`
- `openstack --help`

键入不带任何子命令的 `openstack` 将切换到交互模式，可以在其中键入 `help [subcommand]` 以了解信息。要退出交互模式，请键入 `quit`。

有关 `openstack` 子命令、选项和语法的详细列表，请参见 <http://docs.openstack.org/developer/python-openstackclient/index.html>。

表 1 OpenStackClient 命令等效项

主命令	子命令	openstack 主命令的等效子命令	附注
cinder			
	type-create	volume type create	
	type-key	volume type set/volume type unset	
	service-list	service list	
glance			
	image-create	image create	仅限 API 版本 1
	image-show	image show	
keystone			

主命令	子命令	openstack 主命令的等效子命令	附注
	user-list	user list	
	user-role-add	user role add	
	user-role-list	user role list	
neutron			
	net-create	network create	
	router-create	router create??	
	router-list	router list??	
	router-show	router show??	
	subnet-list	subnet list	
nova			
	flavor-key	flavor set/flavor unset	在 OpenStack Liberty 版本中
	flavor-list	flavor list	
	flavor-show	flavor show	
	image-create	image create	仅限 API 版本 1
	image-list	image list	
	image-show	image show	
	migrate	server migrate	
	resize	server resize	
	service-disable	compute service set/compute service unset	仅限 Keystone 版本 3
	show	server show	

# 索引

---

## A

- 安装 OpenStack
  - 多节点配置, 23
- allowed-mac-address, 102
- allowed-vlan-ids, 102
- archiveadm 命令, 59

## B

- 部署裸机 见 裸机

## C

- 承载 Neutron 的内核区域, 101
- 创建和上载映像的脚本, 62
- 存储节点
  - 关于, 23
  - 多个后端主机, 87
  - 配置, 44
- Cinder
  - NFS 支持, 93
  - SAN 支持, 87
  - ZFSSA 的 iSCSI Cinder 驱动程序, 95
  - 使用 ZFSSA 作为后端存储, 96
  - 可用性区域, 92
  - 在远程主机上设置访问控制列表, 91
  - 备份, 126
  - 安装, 36
  - 用户权限配置文件, 90
  - 目标存储主机, 87
  - 部署选项, 87
  - 配置文件, 36, 45, 88, 97
  - 配置远程存储系统, 87
- Cinder 卷的迁移, 13
- Cinder 用户的权限配置文件, 90
- Cinder 中的 SAN 支持, 87

- Cloudbase-Init, 121
- cloudbase-init 支持, 15

## D

- 单节点 OpenStack 安装
  - 使用 OVM Server for SPARC, 24
- 多节点 OpenStack 安装
  - 3-节点参考体系结构, 23
  - 准备工作, 25

## F

- 风格, 78
  - 参见 VM 实例
    - extra\_specs 属性, 79
    - 修改属性, 79
    - 关于, 68
- 服务管理工具 (Service Management Facility, SMF)
  - 目标主机必需的服务, 91
- 浮动 IP 地址, 72
  - 参见 网络

## G

- Glance
  - 关于, 59
  - 创建和上载脚本, 62
  - 安装, 32
  - 显示映像信息, 61
  - 配置文件, 33

## H

- Heat

- 与 Cloudbase-Init , 121
- 关于, 119
- 安装, 119
- Horizon
  - 查看风格, 79
  - 概述, 66
  - 视频介绍, 68
  - 访问, 65
  - 配置 SSL 访问, 35
  - 项目列表, 67
- I
- Ironic
  - /etc/ironic/ironic.conf 文件, 110
  - AI 配置, 108
  - 使用 UAR 和 IPS, 113
  - 停用节点, 117
  - 安装和配置, 108
  - 服务, 107
  - 组件, 107
  - 置备期间的故障, 117
  - 通过 UAR 部署裸机, 114
  - 部署裸机, 112
- J
- 计算节点, 23
  - 参见 Nova
  - 恢复, 84
  - 配置, 40
- 卷
  - 备份和恢复, 13
  - 迁移, 13
- K
- 可伸缩性, 133
- 控制器节点
  - 关于, 23
  - 配置, 29
- 控制台访问, 43
- 快照, 59
- Keystone
  - sample\_data.sh 脚本, 27
- 安装, 31
- 配置文件, 32
- L
- 裸机, 112
  - 参见 Ironic
  - 与 Ironic 部署在一起, 112
  - 使用 UAR 和 IPS, 113
  - 使用 UAR 进行部署, 114
  - 停用, 117
  - 可配置的节点元素, 112
- L3 代理, 51, 58, 101
- LDoms, 24
- M
- MAC 地址, 动态, 103
- MySQL, 30
- N
- 内存优化, 28
- Neutron
  - L3 代理, 51, 58, 101
  - 使用动态 MAC 地址和 VID, 101
  - 创建路由器, 51
  - 在内核区域中, 101
  - 外部网络, 53
  - 安装, 37
  - 路由器, 51
  - 配置文件, 37
- neutron-l3-agent SMF 服务, 52
- NFS 卷, 93
- Nova
  - 创建 VM 实例, 73
  - 在控制器节点上安装, 34
  - 引导卷, 92
  - 恢复计算节点, 84
  - 显示映像信息, 61
  - 清除, 14, 84
  - 调整大小, 84
  - 迁移实例, 14, 82
  - 配置, 40
  - 配置文件, 34, 40



**O**

OVM Server for SPARC , 24

**P**

平面网络 , 54

**Q**

区域框架 , 40

**S**

升级

    从 Havana , 18

    从 Juno , 18

    注意事项 , 17

实例 见 VM 实例

实例模板 见 风格

实时迁移 , 14 , 82

数据库 , 30

sample\_data.sh 脚本 , 27

SAN 存储 , 13

SC 配置文件 , 59 , 80

SPARC , 24

SQLite , 30

SSH 密钥

    在 EVS 配置中 , 42

    实例迁移 , 83

Swift

    关于 , 24

    配置文件 , 47

**T**

提供商路由器 见 Neutron

统一归档文件

    创建 OpenStack 映像 , 59

弹性虚拟交换机 (Elastic Virtual Switch, EVS)

    evsadm 命令 , 52 , 54

    SSH 密钥 , 42

    配置 , 41

调试选项 , 129

TLS 配置文件 , 36

**U**

user\_reserve\_hint\_pct , 28

**V**

VLAN ID, 动态 , 103

VM 实例 , 73

    参见 Nova

    创建 , 73

    备份 , 60

    密钥对 , 73

    快照 , 59

    挽救 , 60

    映像 , 59 , 67

    添加用户 , 77

    登录 , 76

    调整大小 , 14 , 84

    迁移 , 14 , 82

    风格 , 67 , 68 , 78

**W**

外部网络的路由器 , 51

挽救映像 , 60

网络

    关联浮动 IP 地址 , 72

    内部 , 70

    创建外部网络 , 51 , 54

    创建路由器 , 51

    子网 , 70

    将内部网络连接到外部网络 , 57

网络地址转换 (network address translation, NAT)

    关于 , 51

    禁用安全 NAT , 55

网络时间协议 (Network Time Protocol, NTP)

    客户机 , 30

    客户机配置文件 , 30

    服务器 , 28

    服务器配置文件 , 28

**X**

系统配置文件 见 SC 配置文件

系统信息显示板

- 列出项目, 67
- 创建 VM 实例, 73
- 创建内部网络, 70
- 显示风格, 68, 79
- 查看映像, 67
- 登录, 65
- 视频介绍, 68
- 显示允许的 MAC 地址和 VID, 104
- 项目
  - 修改用户角色, 69
  - 创建, 68
  - 添加用户, 68, 69
  - 缺省, 67
- 虚拟机 (virtual machine, VM) 见 VM 实例
- zonecfg:bootargs 选项, 81

## Y

- 映像, 59
  - 参见 VM 实例
  - 关于, 67
  - 备份, 60
  - 快照, 59
  - 挽救, 60
  - 映像高速缓存, 59
  - 显示信息, 61
  - 注册表服务器, 59
- 硬件的模板 见 风格
- 硬件模板 见 风格
- 远程存储系统, 87
  - 参见 Cinder
  - 启用为目标, 91
  - 必需的 SMF 服务, 91
  - 访问控制列表 (Access Control List, ACL), 91
- 远程访问守护进程 (Remote Access Daemon, RAD), 40
- 云虚拟机 见 VM 实例

## Z

- 注册表服务器, 59
- 租户 见 项目
- ZFS Storage Appliance (ZFSSA), 95
  - 参见 Cinder
  - OpenStack 的后端存储, 95
  - 工作流实用程序, 96